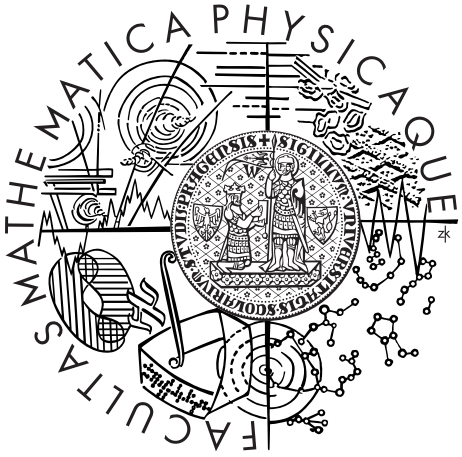
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Miloš Ercegovčević

Joint Learning of Syntax and Semantics

Institute of Formal and Applied Linguistics

Supervisors of the master thesis:	Dr. Ivan Titov Asst. prof. RNDr. Ondřej Bojar Prof. dr. Manfred Pinkal
Study programme:	Informatics
Specialization:	Mathematical Linguistics and Informatics

Prague 2012

Making a statement of gratitude or acknowledgement at a single point in this continuum of ours is an ungrateful endeavor. We risk of being momentarily influenced and deceived by our various limitations. Thus I would like to thank all the great people that had a privilege too meet, hear, interact or even being inspired by. As we jump between our misconceptions we sooner or later became aware of them and greatly appreciate. Thank you!

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Title: Joint Learning of Syntax and Semantics

Author: Miloš Ercegovčević

Department: Department of Computational Linguistics

Supervisors: Dr. Ivan Titov, Asst. prof. RNDr. Ondřej Bojar, Prof. dr. Manfred Pinkal

Abstract: This master thesis addresses the problem of learning latent levels of abstraction of shallow semantics. We break assumptions made in the annotation of semantic resources that aim at providing a fixed number of semantic roles (e.g. PropBank) and furthermore learn varying levels of abstraction of key linguistic constructs: semantic frame, verb, lexical and syntactic classes. By implementing our model in terms of latent grammars, our induced structures perform comparably with state-of-the-art models in semantic role labeling across multiple languages. Moreover, we show close resemblance of the assumed linguistic properties with the abstractions found in FrameNet. The final outcome is a language-independent, feature-less model of semantic information with meaningful structures and empirically validated performance.

Keywords: semantics, syntax, joint learning, latent variables, language–independent

Název práce: Společná učení syntaxí a sémantiky

Autor: Miloš Ercegovčević

Katedra: Ústav formalní a aplikované lingvistiky

Vedoucí diplomové práce: Dr. Ivan Titov, Asst. prof. RNDr. Ondřej Bojar, Prof. dr. Manfred Pinkal

Abstract: Diplomová práce se zabývá problémem strojového učení nepozorovaných úrovní abstrakce mělké sémantické reprezentace. Odstraňujeme předpoklady, které se při sémantické anotaci lingvistických zdrojů obvykle činí, např. pevný počet sémantických rolí v PropBanku, a učíme se klíčové lingvistické prvky této anotace (sémantické rámce, slovesa, lexikální a syntaktické třídy) s různou mírou abstrakce. Model implementujeme pomocí latentních gramatik a získané struktury je možné použít pro úlohu značkování sémantických rolí (semantic role labeling, SRL) v několika jazycích s přesností srovnatelnou s jinými současnými přístupy. Navíc ukazujeme, že tyto struktury jsou velmi blízké abstrakcím, které je možné pozorovat ve FrameNetu. Celkovým výsledkem je tak jazykově-nezávislý model sémantické informace bez rysů, který produkuje interpretovatelné struktury a jeho použitelnost je na úloze SRL empiricky ověřena.

Klíčková slova: sémantika, syntaxe, joint learning, latentní proměnné, jazyková nezávislost

Contents

1. Introduction

1.1 Semantics in Natural Language Processing

Empiricist methods are dominant in the field of Computational Linguistics, ranging from simple tasks, such as part-of-speech speech tagging, chunking, named entity recognition to more complex tasks like syntactic parsing, speech recognition or machine translation. After huge improvements in stochastic parsing of natural languages, the field has begun to impose tasks that involve a higher level of abstraction such as ***semantic parsing***, going toward semantic understanding. Even though characterizing *who* did *what* to *whom*, *where*, *when*, and *how* might not solve the long–reaching goals of Artificial Intelligence, the task had some successful application domains such as information extraction [29], question answering [28], textual entailment [26] and machine translation [33][15]. Starting from purely supervised approaches to semantic parsing, recent research also shows quite promising results in unsupervised methods. However, several fundamental questions remain open even in this shallow form of semantic parsing. Namely, levels of abstraction have been de facto imposed by annotated corpora such as FrameNet [2] and Propbank [22], which in turn have been shown to be very limiting in their out-of-domain performance [30]. Furthermore, state–of-the–art performance is achieved by feature engineering [9], which is a very tedious and time–consuming task, one which usually does not scale neither domain–wise nor language–wise. Going toward the highly ambitious goal of defining direct correspondences between natural languages on the semantic level which will undoubtedly have to tackle those problems. While unsupervised approaches have tried to resolve some of the problems above, their performance in even simple tasks such as part–of–speech tagging is questionable [5]. This master thesis will try to tackle learning of ***latent semantic representations*** in a semi–supervised setting with ***varying levels of abstraction***, by jointly learning syntactic and semantic dependencies and evaluating them on data provided by $CoNLL09$ shared task [9]. Specifically, by ***breaking assumptions*** made in corpora annotation of multilingual data provided by $CoNLL09$ we learn appropriate representations for semantic roles, their lexical and syntactic elaboration and assumed general frame classes specified with linking alternations by incorporating latent variables in the model, all with a goal of being as predictive as possible for the semantics (i.e. semantic roles).

1.2 Modeling Semantics

The picture that had emerged in previous years is that corpora annotations driven by linguistic consensus are not the most representative for explaining underlying linguistic phenomena. Their failures are evident in self–representation – in learning for the same level of analysis (i.e. clear supervised setting) as well as prediction – using them as an intermediate representation (i.e. as features). Various approaches have been devised to tackle this ***un–representativeness*** of the human derived corpora annotations [23], mostly employing discriminative machine learning approaches. However, generative approaches in general have shown to

be better on a lower scale [20] and have a nice convenient property that they can expressively handle multiple levels of latent variables in the model. Exploiting recent successes in Bayesian modeling with hidden variables [17], in this master thesis we use a generative latent variable model to tackle ***joint learning of syntax and semantics***. Following recent practices, we assume syntax as represented by dependency trees to be provided to us [30][24] and focus on learning semantics as represented by semantic roles. Crucially, our model learns appropriate levels of abstraction for both syntax and semantics in a joint model, while in the inference phase it predicts level of syntactic analysis most appropriate for the overall semantic representation. Assuming exact number of semantic roles and predicate fillers has been assumed so far in context of *ProbBank* [22] and learning them has not been attempted. However, starting from the *FrameNet* intuition of ***the hierarchy of the semantic frames*** we treat learning of semantic abstraction as hidden information, which should maximize the likelihood of training data. ***Role fillers*** have been assumed as verb-specific in corpus annotations, as in *PropBank*, but simple investigation shows that they share a lot of lexical or syntactic similarity. Furthermore, role fillers generalize among themselves with inhibition of different lexical information; we model such a behavior with latent lexical categories which can serve as word class information. Nontrivial ***interaction between syntactic dependencies and semantic roles*** is as well captured with latent variables, similarly as in recent success in unsupervised semantic role induction [32]. The de facto linguistic background of jointly learning syntax and semantics is driven by the ***linguistic theory of linking*** [1]. Linking theory [14] implies that syntactic behavior can be determined from the underlying semantics. We model linking alternations jointly in our model with latent frames, semantics, syntax and lexical categories. Such an approach can be seen as both supervised, as the backbone structures are provided to us, and unsupervised because the model softly clusters the observed and unobserved variables into statistically dependent groups, resulting in what one may call ***semi–supervised learning*** [2].

1.3 Road Map

Chapter 2 introduces necessary theoretical and practical properties on treating semantics in the field of Computational Linguistics. We explore two most common annotation schemata *PropBank* and *FrameNet*. In chapter 4 we briefly point to computational treatment of supervised and unsupervised approaches on the task of semantic role labeling. Chapter 5 introduces our key scientific framework of Latent Probabilistic Context–free Grammars for modeling semantics. Finally in chapter 6 we tackle the problem from both modeling and technical perspectives. We present our latent variable model that without any features automatically learns appropriate representations and perform well on the task of semantic role labeling. Chapters 6 and 7 comment on qualitative and quantitative results, and future work respectively.

[1] See Chapter 2 for details.

[2] Please note that term semi–supervised in our model comes from exploiting unlabeled structures rather than unlabeled data.

2. Semantic theory and Computational Resources

2.1 From linguistic theory to computational practice

Semantic analysis of sentence-level utterances aims at characterizing events and their participants. The event is activated by the event invoker that characterizes *what* took place, and further specifies the *who* and *whom* has the processes undergone and some general properties like *where* or *when* [16]. The event by itself is a carrier of the information and it is most usually represented by a predicate, while the participants and properties define roles with respect to the predicate.

Consider for example the following sentence [1]:

- $[\text{The girl on the swing}]_{\text{Agent}}\ [whispered]_{\text{Pred}}\ \text{to}\ [\text{white boy beside her}]_{\text{Recipient}}$.

Defining this example from some level of abstraction, we can say that the Conversation event is invoked by the predicate *whispered* and that the participant *the girl on the swing* is the agent of the event while *the boybeside her* is the patient.

The theory of semantic roles goes far as thousands of years in Panini's Karaka theory. The whole ***spectrum of generality*** of the roles has been defined in theory as well as in practice [7]. At one end of the spectrum, there are specific roles such as $FromAirport$, $ToAirport$ or $Depart$ that found useful application in natural language understanding specifically in dialog systems. On the other end, there are more coarse–grained roles that can be merged down to as few as two roles (e.g. $Proto-agent$ and $Proto-patient$) to several roles such as Fillmore's list of nine: $Agent$, $Experiencer$, $Instrument$, $Object$, $Source$, $Goal$, $Location$, $Time$, and $Path$. The more general roles have been proposed by the linguists who are more interested in describing generalizations across syntactic realizations of their arguments as driven linguistic theory of linking. On the other hand, computer scientists have been proposing more specific roles as they are more interested in details of the realization of the arguments.

2.1.1 Linking Theory

Linking theory [14] argues that the alternation behavior of the verb as described by the syntactic frames is a direct reflection of the underlying semantics. The theory introduces Levin verb classes, which are defined by the syntactic frames which respectively constrain allowable arguments of semantics. Thus a ***verb class*** is defined as the possibility of a particular verb to occur in pairs of syntactic frames. It is further argued that the syntactic frames are meaning-preserving and

[1] Example taken from [16] .

that classes tend to share some of the semantic behavior; the principle is called ***diathesis alternations***.

For example, let us consider alternations of break verbs *break*, *shatter* and *smash*. All of them can be characterized in their ability to occur in the middle construction like in [2]:

- Glass *breaks/shatters/smashes* easily.

Where the middle construction represents a special case of intransitive construction [1]. Now consider the verb *cut* which is very similar to the verbs above and also tends to occur in the middle construction, like in [2]:

- John *cut* the bread.

However *cut* cannot occur in the intransitive construction like in *The bread cut*, while *The window broke* is very plausible. On the other hand *cut* can occur in the conative like in[2]:

- John valiantly *cut* at the frozen loaf, but his knife was too dull to make a dent in it.

This kind of behavior is unusual for *break* verbs, because *cut* is a change–of–state verb that describes series of actions, while *break* verbs only specify the resulting state of action.

2.1.2 FrameNet

FrameNet proposes roles that lie on the spectrum of generality somewhere between extremely specific roles (e.g. like in our airport example) and extremely general roles (e.g. $Proto - Agent$ and $Proto - Patient$) [2]. The basic concept is that of the frame. A ***frame*** is a schematic representation of situations that involve various participants, props, and other conceptual roles. For example, the frame *Probability* [3], shown below, is invoked by the semantically related nouns *chance*, *chances*, *likelihood*, *odds*, *probability*, *significance*, and is defined as follows:

- *This frame characterizes the likelihood that a* ***Hypothetical_event*** *will happen as a position on a scale of impossible to inevitable. The likelihood can expressed as numerical* ***Odds*** *or a metaphorical representation of the* ***Position*** *on a scale*

Roles defined by this frame are *Hypothetical_event*, *Odds* and *Position*. With the following interpretation:

- ***HYPOTHETICAL_EVENT*** *The event that is expected to happen with a certain likelihood. He's got a small* ***chance of making it out alive***.

[2]Example taken from [22] .

[3]All examples from this section can be *FrameNet* can be found at `https://framenet.icsi.berkeley.edu/fndrupal/`.

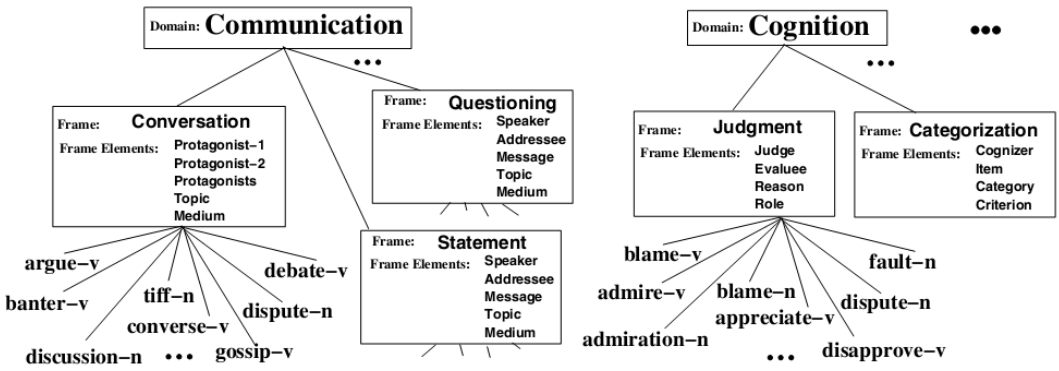


Figure 2.1: Domains of *FrameNet* defined on varying levels of abstraction.

- ***ODDS*** *A numerical representation of the probability that a Hypothetical_event will occur. There's a* ***95 % chance*** *of rain today.*
- ***POSITION*** *A metaphorical representation of the position on the scale of likelihood that the Hypothetical_event will occur.* ***Chances*** *are* ***slim*** *that he'll pull through.*

The roles are defined on a frame basis and are shared by all lexical entries belonging to the frame. The diversity of the following example sentences for the *Probability* frame demonstrate the broad semantic coverage of *FrameNet*:

- Generally $[\text{less}]_{\text{Position}}$ $[chance]_{\text{chance}.n}$ [of temporal variations $]_{H\ \text{event}}$
- $[chances]_{\text{chance}.n}$ are [I attacked the other books too$]_{H_{\text{event}}}$

This annotation clearly shows the ***level of generality*** of semantic frames defined by *FrameNet*. On the one hand, it is specific enough to capture lexical and syntactic information and on the other hand general enough to talk about abstract notions of an inheritance hierarchy of semantic frames. Indeed, *FrameNet* allows generalizations across different categories of verbs, nouns, and adjectives with each of them adding semantics to the general frame or highlighting a particular aspect of the frame. Conversely, many of the phenomena in the methodology of *FrameNet* remain problematic. For example, it is clear that there is no always a direct correspondence between syntax and semantics. Further, the development methodology of *FrameNet* has a big impact on what researchers should expect in practical applications. In the first step, a set of semantic frames was chosen for the general domains of interests (see Figure 2.1) [4]. Consequently, a list of target words was compiled for each frame and example sentences were chooses by searching the list of candidates in *British National Corpus*. Various patterns over lexical items and part–of–speech sequences in the target words' context were performed and the example for annotation chosen with the aim of coverage. Finally, sentences were manually annotated and checked for consistency. It is clear that such an approach emphasizes completeness of examples for ***lexicographic needs*** rather real word distribution of semantic phenomena.

[4]Figure taken from [7] .

Table 2.1	
Subtypes of the ArgM modifier tag	
LOC: location	*CAU: cause*
EXT: extent	*TMP: time*
DIS: discourse connectives	*PNC: purpose*
ADV: general-purpose	*MNR: manner*
NEG: negation marker	*DIR: direction*
MOD: modal verb	

Table 2.1: Adjunct semantic roles defined by the PropBank.

Frameset accept.01 *take willingly*
Arg0: Acceptor
Arg1: Thing accepted
Arg2: Accepted-from
Arg3: Attribute

$[\text{He}]_{\text{Arg0}}\,[\text{would}]_{\text{ArgM}-\text{MOD}}\,[n't]_{\text{ArgM}-\text{NEG}}\,\textit{accept}\,[\text{anything of value}]_{\text{Arg1}}\,[\text{from them}]_{\text{Arg2}}$

Frameset kick.01 *drive or impel with the foot*
Arg0: Kicker
Arg1: Thing kicked
Arg2: Instrument (defaults to foot)

$[\text{John}_i]_{\text{Arg0}}\,\textit{tried}\,[\text{*trace*}_i]_{\text{Arg0}}\;[\text{the football}_i]_{\text{Arg1}}\,.$

Figure 2.2: Sample Framesets as defined by the *PropBank*.

2.1.3 PropBank

The issues with the broad-coverage and statistically unrepresentative samples of *FrameNet* are partially solved by schemata defined in ***PropBank*** corpus. Taking into account that the other end of spectrum (i.e. defining a small set of universal roles) is difficult, the roles are annotated on ***a verb per verb basis*** [22]. Individual semantic roles of a predicate are numbered starting from 0. Given a particular verb, $A0$ is most probably the argument that exhibits features of a prototypical agent (i.e. $Proto-agent$) while $A1$ is a prototypical patient or theme (i.e. $Proto-patient$).

Further, developers point out that ***no consistent generalizations*** can be made across verbs for higher numbered arguments [22]. Claims go further to that the effort was made to define roles consistent with respect to the roles across members of *VerbNet* classes [27]. In addition to these core roles, more general roles that can apply to any verb were defined – adjunct roles (check Table 2.1).

Frameset: decline.01 *go down incrementally*
Arg1: entity going down
Arg2: amount gone down by, EXT
Arg3: start point
Arg4: end point
... [its net income]$_{\text{Arg1}}$ *declining* [42%]$_{\text{Arg2-EXT}}$ [to \$121million in the first 9 months of 1989]$_{\text{ArgM-TMP}}$.

Frameset: decline.02 *demure, reject*
Arg0: agent
Arg1: rejected thing
[A spokesman]$_{\text{Arg0}}$ *declined* [$*$trace $*$ to elaborate]$_{\text{Arg2-EXT}}$

Figure 2.3: Defining verb meaning by the number of verb's arguments.

Frameset open.01 *cause to open*
Arg0: agent
Arg1: thing opened
Arg2: instrument
[John]$_{\text{Arg0}}$ *opened* [the door.]$_{\text{Arg1}}$
[The door]$_{\text{Arg0}}$ *opened*.
[John]$_{\text{Arg0}}$ *opened* [the door]$_{\text{Arg1}}$ [with his foot.]$_{\text{Arg2}}$

Figure 2.4: Sentences with transitive and intransitive uses of the verb *open*.

Distinct usages of a verb are captured by the set of its semantic roles, which is called a ***Roleset***. The *Roleset* can be associated with the set of syntactic frames that suggest allowable syntactic variations with respect to that set of roles and jointly they constitute a *Frameset*. Consequently, polysemous verbs may have more than one *Frameset* as represented by the defined differences in meaning. Figure 2.2 shows sample *Framesets* [5].

Discriminative criteria for distinguishing framesets are based on both syntax and semantics. For example, two verb meanings are different if they take a different number of arguments (see Figure 2.3) [5].

Furthermore, verbs which do preserve the meaning with an alternation such as causative/inchoative or object deletion belong to the same frameset, while allowing for the case in which some arguments could be left unspecified. Such

[5]Example taken from [22] .

Frameset see.01 *view*
Arg0: viewer Arg1: thing viewed
$[\text{John}]_{\text{Arg0}}\ saw\ [\text{the President.}]_{\text{Arg1}}$ $[\text{John}]_{\text{Arg0}}\ saw\ [\text{the President collapse.}]_{\text{Arg1}}$

Figure 2.5: An example of a syntactic misleading for *FrameSet* identification.

are the examples for transitive and intransitive uses of the verb *open* as depicted in Figure 2.4 [5].

Finally, as with any system of rules, the syntactic type of the arguments does not constitute the criterion for distinguishing between framesets where both an *NP object* or a *clause object* satisfy the constrains (e.g. see Figure 2.5) [5].

3. Computational modeling of semantics

In this chapter we explore current computational approaches to the problem of semantic role labeling. In Section 3.1 we briefly discuss current ***supervised*** state–of–the–art approaches emphasizing their most relevant properties. Further, most relevant ***unsupervised*** approaches are explored in Section 3.2 with overall intent to reduce the conceptual modeling gap between them and the supervised setting for which we are after.

3.1 Supervised learning

Supervised semantic role labeling is usually captured with the following subtasks:

- ***Argument identification*** : identifying the boundaries of arguments of a predicate and
- ***Argument classification*** : labeling them with semantic roles.

As arguments of a predicate can be continuous or discontinuous sequences of words, any subsequence of words in a sentence is an *argument candidate*. The $argument\ identification$ has been usually tackled with heuristics or by training discriminative classifiers [1]. Then the task of $argument\ classification$ is to assigns semantic labels to the detected argument candidates by usually using feature–rich classifiers [9][16]. We depict this standard pipeline in Figure 3.1. For example, the state–of–the–art system [4] on *Chinese*, *Czech*, *English* and *German* is using a pipeline of independent, local classifiers that identify the predicate's sense, the arguments of the predicate, and the argument labels.

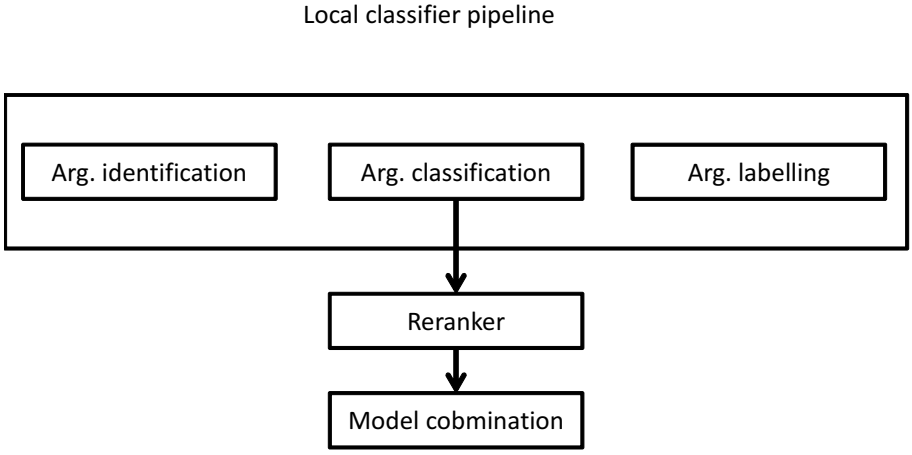


Figure 3.1: The standard supervised architecture for semantic role labelling.

[1] Note that argument identifier can be trained on a substantially smaller portion of the training data and still have high performance.

Further, a system generated with a beam search a set of candidates which were then re–ranked using a joint learning approach that combined local models and propositional features. Finally, feature selection was done which moderately improved the performance.

Because the full technical specification and the description of state-of-the-art systems would be highly diverse and in some way specific to the machine learning framework that has been particularly used we will instead comment on the general architectures, drawbacks, complexity involved and most important properties. Further, in Chapter 5, we will argue about the shortcomings of these methods. The reader interested in details is encouraged to read some of the state–of–the–art research [34] [4].

One of the most important properties of discriminative models is in their ***features set***, which is used in the various processing steps in the model. We depict one of these feature sets in Figure 3.2. As you can observe, multitude of lexical and structured information has been observed as being predictive with respect to semantic roles by this discriminative system. Thus we argue that the property of the utmost importance is the ***exclusive selection*** of discriminative models toward some subset of features which in turns out to be very much different across languages. This property is in much of a contradiction with some of the language modeling phenomena as derived by linguistic theories and by mere nature of the language processing as being highly uncertain domain of interest. First, as the most of the linguistic theories argue, the correspondence between various levels of linguistic abstraction is observable and relevant across multiple languages, as the recent success in unsupervised multilingual parsing is indicating [19]. And second, as we do not know these abstractions and correspondences we should appropriately reason over them (i.e. try to incorporate their learning in the model)and should not to selectively exclude or include them. Thus we argue by using the exclusive set of features as the one in the table 3.2 one imposes two kinds of domain over–fitting:

- ***Lexical specificity*** : defined as over–reliance on the particular properties of the surface structure which are most probably genre–related
- ***Structured specificity*** : defined as over–reliance on the particular properties of intermediate linguistic structures which are most probably error–driven [2] and genre–related

Further, important property that has been shown successful in previous works is to employ ***structural and linguistic constraints*** into the semantic role labeling [25]. The model devised on these constructs has following setup: given a predicate, system treated all possible spans as candidate arguments and at the first stage of pruning, which was done using a syntactic parse trees, a model deterministically filtered the space. This was followed by filtering with classifiers that did argument identification and an argument classification. In the final stage, all labeled arguments with their posterior probability and a set of linguistically and

[2]Error–driven in the sense that a human cannot possibly create a fully adequate theory in terms of explanatory adequacy and even the ones that are created can suffer from inter–annotator disagreement on the annotated data.

	Argument identification							Argument classification						
	ca	ch	cz	en	ge	ja	sp	ca	ch	cz	en	ge	ja	sp
PredWord									•		N		•	
PredPOS				N	•				•		V		•	
PredLemma				N	•			•	•	•	N,V	•		•
PredDeprel														
Sense		•	•	V	•			•	•	•	N,V	•	•	•
PredFeats			•							•		•		
PredParentWord				V					•		V		•	
PredParentPOS				V							V	•		
PredParentFeats			•											
DepSubCat	•	•												
ChildDepSet	•				•			•	•		V	•	•	•
ChildWordSet				N						•			•	
ChildPOSSet		•							•		N		•	
ArgWord	•	•		N,V	•	•	•	•	•	•	N,V	•	•	•
ArgPOS	•	•		N,V	•	•	•	•	•	•	N,V		•	
ArgFeats	•							•		•		•		•
ArgDeprel	•	•	•	V	•		•	•	•	•	V	•		•
DeprelPath	•	•	•	N,V	•	•	•	•	•		V	•		
POSPath	•	•	•	N,V	•	•	•			•	V	•	•	
Position			•	N,V		•		•	•	•	N,V		•	•
LeftWord	•				•			•	•		N		•	•
LeftPOS						•			•		V			
LeftFeats						•	•					•		
RightWord	•			N				•	•		N,V			•
RightPOS				N				•	•		N,V			•
RightFeats								•						•
LeftSiblingWord	•						•	•	•		N		•	
LeftSiblingPOS					•	•		•	•		N,V		•	
LeftSiblingFeats			•					•				•		
RightSiblingWord	•	•		V	•	•	•		•			•		•
RightSiblingPOS												•	•	
RightSiblingFeats													•	

Figure 3.2: Final feature set obtained by the system *Nugues* [4].

structurally motivated constraints were submitted as an integer linear program in order to make a globally consistent prediction. Constraints have been devised with structural or linguistic properties: arguments cannot overlap with the predicate; arguments cannot exclusively overlap with the clauses; if a predicate is outside a clause, its arguments cannot be embedded in that clause and many more. With this paradigm, discriminative models enabled leveraging linguistic and structural prior knowledge directly for achieving high performance. Most of the state-of-the-art systems use this technique on the top of local classifiers [34] [4].

However, previous work also shows approaches where only ***minimal feature engineering*** was used and thus appropriate features for a task at question were learned automatically. One of these approaches [6] use incremental parsing model with synchronous syntactic and semantic derivations. The derivations are modeled using latent variables in terms of Incremental Sigmoid Belief Networks [10] and in that way enable induction of shared features for both syntax and semantics. Technically, a model has one input queue with two stacks that models derivations as synchronized at each input word. The whole model is language independent and reaches state-of-the-art performance on *CoNNL09* task.

3.2 Unsupervised learning

The first fully unsupervised system [8] for semantic role labeling aimed at devising a ***broad–coverage language lexical resource***. The model was intended to learn ***verbs behavior*** that can be easily extended to new text genres and languages. Specifically, the model related a verb, its semantic roles and their possible syntactic alternations. Syntax was not modeled but gained from corpora annotation with automatic parsers and translated into a fairly language–independent set of syntactic relations, a subset form of a dependency grammar. Furthermore, a simplistic representation of semantics was devised which only had five core arguments (similar to *PropBank*) and one adjunct role which was shared across all verbs. The task of argument identification was simplistically tackled and assumed that arguments are direct dependents of the verb in the syntactic representation. Table 3.2 offers an illustration of the process of extracting semantic representation.

A deeper market plunge today
could *give* them thier first test.

Verb: give

Syntactic Relation	Semantic Role	Head Word
subj	ARG0	plunge/NN
np	ARGM	today/NN
np#1	ARG2	they/PRP
np#2	ARG1	test/NN

$$v = \text{give}$$

$$l = \{\text{ARG0} \rightarrow \text{subj}, \text{ARG1} \rightarrow \text{np\#2}, \text{ARG2} \rightarrow \text{np\#1}\}$$
$$o = [(\text{ARG0}, \text{subj}), (\text{ARGM},?), (\text{ARG1},\text{np\#1}), (\text{ARG2},\text{np\#2})].$$

$$(g_1, r_1, w_1) = (\text{subj}, \text{ARG0}, \text{plunge/NN})$$
$$(g_2, r_2, w_2) = (\text{np}, \text{ARGM}, \text{today/NN})$$
$$(g_3, r_3, w_3) = (\text{np1}, \text{ARG\#2}, \text{they/PRP})$$
$$(g_4, r_4, w_4) = (\text{np2}, \text{ARG\#1}, \text{test/NN})$$

Table 3.2: Example how the model extracts semantic roles and relates them to the syntax and surface forms.

Thus provided a verb *give* and its four direct dependents the syntax was unambiguously stripped off from dependency representation – *subj*, *np*, *np*#1, *np*#2. Then task was in ***discovering the mapping*** (i.e. *linking*) between observed syntactic dependencies and unobserved semantic roles. Furthermore, the l unordered set (i.e. *linking* set) specified the mapping for core roles and the

o ordered set further implied surface ordering and addition of adjunct roles (e.g. $ARGM$ in the second position in Table 3.2).

The graphical representation of the model is depicted in Figure 3.3. The model defines a joint probability distribution over elements of a single verb instance: verb lemma, syntactic dependencies, semantic roles, linking and head words. The model first generates a verb – v and then, conditioned on the choice of the verb, it generates a linking – l which in turn defines a set of core semantic roles – r and the syntactic relations – g that express them. One possible drawback with this kind of a model is that the linking is specified only for core semantic roles and the process introduces uncertainty about the choice of linking and its representation in the ordered list. Consequently, an additional variable (i.e. o variable) had to be introduced in order to capture this uncertainty, which in fact increased the complexity of the model. Finally, each of roles generates its surface form – w concatenated with a pos tag.

This model is not only important as the first fully unsupervised system but also as the first system which directly modeled ***linking alternations*** in the compact form. In some point of view the model tries to learn constructs that might correspond to the $Framesets$ in terms of $PropBank$.

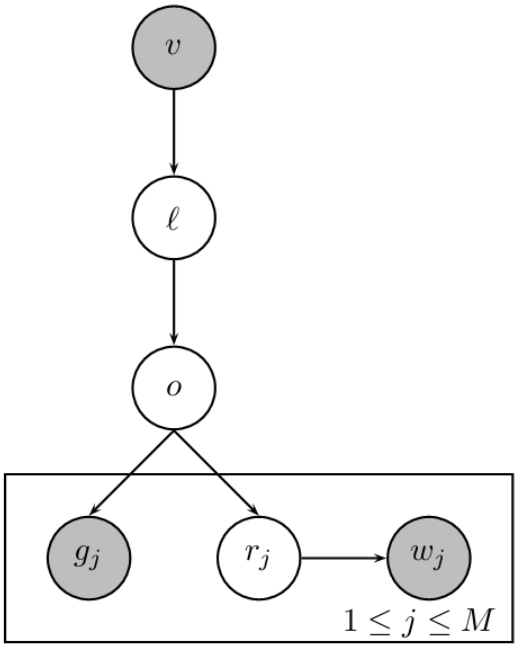


Figure 3.3: Graphical model of the first fully unsupervised system [8] for semantic role labeling.

Further, recent work has also shown that ***clustering of predicates*** can be beneficial to the task at question. Titov and Klementiev [31] have explored this kind of an approach while learning semantic roles in a unsupervised setting for the task of question answering.

4. Computational modeling of uncertainty

In this chapter we explore our main mathematical framework that we will use for modeling semantic role labeling – Latent Probabilistic Context–free Grammars (PCFG–LA). In the following chapters the semantic role labeling with PCFG–LA will be provided in more detail and in this chapter we focus more on formal definitions of the framework itself.

4.1 Latent Probabilistic Context Free Grammars

PCFG–LA is a ***generative*** model of parse trees. The observed nonterminal symbols correspond to parse trees and are treated as ***incomplete*** data. When each observed nonterminal symbol T gets labeled (***clustered***) with the latent variable assignment, the resulting symbol $T[X]$ is completely observed. PCFG–LA thus provides further (unsupervised) ***refinement*** of the grammar captured in the treebank.

For example consider the pair of observed and unobserved nonterminal symbols shown in Figure 4.1. When each of the nonterminals in the constituency tree (depicted as the right tree) gets annotated with latent annotation x we get completely observed tree (depicted as the left tree). Furthermore, by observing initial constituency tree as incomplete data, ***expressive power is increased*** and consequently some of the formal assumption properties of Context–free Grammars are relaxed in the PCFG–LA (i.e. *context–free*: productions are independent from the neighboring nodes, *ancestor–free*: productions are independent from the ancestor's node).

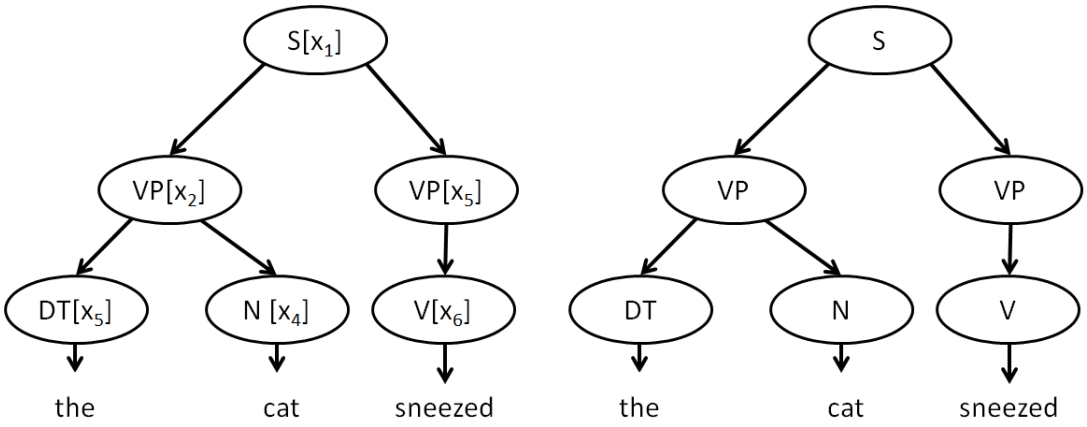


Figure 4.1: Completely and incompletely observed constituency trees.

A grammar that generates complete parse trees is generated exactly as in Context–free Grammar with an the exception that every observed node has to be specified (***clustered***) with a latent annotation symbol.

We use the formulations from [17] and [23]. Formally, PCFG-LA G as is a tuple $G = \langle N_{\text{nt}}, N_t, H, R, \pi, \beta \rangle$, where:

- N_{nt} is a set of observable symbols
- N_t is a set of terminal symbols
- H is a set of latent variable symbols
- R is a set of observable CFG rules in *Chomsky Normal Form* [1]
- $\pi(x)$ is the probability of the root taking assignment x
- $\beta(r)$ is the rule probability of the rule r

The probability of the complete parse tree:

$$P\left(T\left[X\right]\right) = \pi\left(S\left[x_1\right]\right) \prod_{r \in D_{t[X]}} \beta\left(r\right)$$

where $\pi\left(S\left[x_1\right]\right)$ is the probability of generating $S\left[x_1\right]$ as the root symbol, $D_{t[X]}$ denotes the multiset of annotated CFG rules used in the generation of $t[X]$ and $\beta(r)$ the is the probability of the rule r. Thus the probability of the complete parse tree from Figure 4.1 is defined as:

$$\begin{gathered} P\left(T\left[X\right]\right) = \pi\left(S\left[x_1\right]\right) \times \beta\left(S\left[x_2\right] \rightarrow NP\left[x_2\right] VP\left[x_5\right]\right) \times \\ \beta\left(NP\left[x_2\right] \rightarrow DT\left[x_3\right] N\left[x_4\right]\right) \times \beta\left(DT\left[x_c\right] \rightarrow the\right) \times \beta\left(N\left[x_4\right] \rightarrow cat\right) \times \\ \beta\left(VP\left[x_5\right] \rightarrow V\left[x_6\right]\right) \times \beta\left(V\left[x_6\right] \rightarrow sneezed\right) \end{gathered}$$

Furthermore, the probability of the observed tree $T\left[X\right]$ is gained by summing out the latent annotation symbols X:

$$P\left(T\right) = \sum_{x \in H^m} \pi\left(A_1\left[x_1\right]\right) \prod_{r \in D_{T[X]}} \beta\left(r\right) = \sum_{x_1 \in H} \sum_{x_2 \in H} \cdots \sum_{x_n \in H} P\left(T\left[X\right]\right)$$

where $X = (x_1, x_2, \ldots, x_n) \in H^m$ is a vector of latent annotation symbols and x_i is the latent annotation symbol attached to the i–th nonterminal node. The equation has the exponential cost. The reason being that calculation at node n has a cost that exponentially grows with the number of $n's$ daughters because the summation involves $|H|^{d+1}$ combination of latent variables assuming that n had d daughters. However this equation can be computed using dynamic programming methods.

For this purpose, we need definitions of ***forward*** and ***backward*** probabilities in the context of PCFG–LA. Thus given a sentence $w_1 w_2 ... w_n$ and its corresponding parse tree T backward probabilities $b_T^i\left(x\right)$ are computed for the i–th nonterminal node and for each $x \in H$ as:

[1] A Context–free Grammar is in Chomsky Normal Form if all its production rules are at most binary. Thus we assume appropriate binarization for the raw treebank trees.

- If node i is a preterminal node above a terminal symbol w_j:

$$b_T^i(x) = \beta(N_i[x] \to w_j)$$

- Otherwise, let j and k be the two daughters of the node i then:

$$b_T^i(x) = \sum_{x_j, x_k \in H} \beta(N_i[x] \to N_j[x_j] N_k[x_k]) \times b_T^j(x_j) b_T^k(x_k)$$

where $N_i \in N_T$ is the nonterminal label of the i–th node. Then the probability of an observed tree is:

$$P(T) = \sum_{x_k \in H} \pi(N[x_1]) b_T^i(x_1)$$

And similarly the forward probabilities $f_T^i(x_1)$:

- If node i is the root node:

$$f_T^i(x) = \pi(N[x_1])$$

- Otherwise, let j be the right sibling of i and k its mother:

$$f_T^i(x) = \sum_{x_j, x_k \in H} \beta(N_j[x_j] \to N_i[x_j] N_k[x_k]) \times f_T^j(x_j) b_T^k(x_k)$$

Provided annotated corpora of observable trees $T = \{T_1, T_2, ..., T_k\}$, where $N_1^i, \ldots, N_{m_i}^i$ are the labels of nonterminal nodes in T_i, we can estimate parameters $\theta = (\beta, \pi)$ with ***EM algorithm***. Derivation of the EM is similar as for other latent variable models and it is defined as a constraint optimization problem:

$$Q(\theta'|\theta) = \sum_{T_i \in T} \sum_{x_i \in H^{m_i}} P_\theta(X_i|T_i) \log P_{\theta'}(T_i[X_i])$$

which iteratively updates the values of the parameters θ and θ' for the probability distributions P_θ and $P_{\theta'}$ respectively; and where $P(X|T) = P(T[X]) / P(T)$ is the conditional probability of latent annotations given an observed tree T. By incorporating Lagrange multiplier method and re–arranging the results using the backward and forward probabilities one obtains the update formulas. For the detailed formalae description and furher details please see [17].

Given learned parameters θ, labeling a new sentence w can be formulated as:

$$T_{\text{best}} = \underset{T \in C(w)}{\text{argmax}} P\left(T|w\right) = \underset{T \in C(w)}{\text{argmax}} P\left(T\right)$$

where $T \in C\left(w\right)$ is a set of all possible parses of w under observable grammar. The expression above involves so called sum-of-product calculation which can be proved ***intractable*** (NP-hard) for latent variable models. There are few approximations over posterior marginal of the parse tree distribution but here we present ***Max-Rule-Product*** objective which is one of the most used one in the context of PCFG–LA:

$$T_G = \underset{T}{\text{argmax}} \prod_{e \in T} q\left(e\right); q\left(A \rightarrow B, C, i, k, j\right) = \frac{r\left(A \rightarrow B, C, i, k, j\right)}{P_{\text{IN}}\left(\text{root}, 0, n\right)}$$

which selects the tree that has greatest chance of having all rules correct with the assumption that the correctness of all rules are conditionally independent. And its rule score:

$$r\left(A \rightarrow B, C, i, k, j\right) = \sum_x \sum_y \sum_z P_{\text{OUT}}\left(A\left[x\right], i, j\right) P_{\text{IN}}\left(B\left[y\right], i, k\right) P_{\text{IN}}\left(C\left[z\right], k, j\right)$$

where:

$$P_{\text{OUT}}\left(A\left[x\right], i, j\right) = P\left(w_{1:i} A\left[x\right] w_{j:n}\right)$$

$$P_{\text{IN}}\left(B\left[x\right], i, k\right) = P\left(w_{i:k} | B\left[x\right]\right); P_{\text{IN}}\left(C\left[x\right], k, j\right) = P\left(w_{k:j} | C\left[x\right]\right)$$

and where A, B, C are nonterminals, x, y, z are latent annotation symbols and i, j, k are between word indices.

When deciding on the number of latent annotation symbols, one usually uses a fixed number of symbols for each nonterminal symbol. However, that approach has been shown to lead to oversplitting and thus faster overfitting of training data and unmanageable growth of the grammar. [23] tackle this problem by incorporating a ***split-merge formulation***. Specifically, all latent variables are first split in two and then only the ones that gave the highest improvement in likelihood are kept, while others are merged back to the state of the previous iteration. For the reader interested in the details, we suggest [23].

5. Modeling Semantics: Probabilistic latent variable approach

In making a model for computational processing of linguistic structures, one has to delve deeper into the specifics of the ***underlying problem***. In our opinion, current approaches to semantic role labeling, at least in the domain of *Propbank*, have largely ignored that important question. As we discussed in Section 3, supervised approaches rely to a huge extent on the following factors:

- availability of sufficient amount of ***annotated data***
- existence of a well-defined ***set of features*** relevant to the task
- assumptions about the correctness of the underlying ***linguistic structures***

5.1 Key insights

First of all, the availability of a sufficient amount of annotated data is true only for some languages. And in that case, the amount of data required to make appropriate generalizations might not be sufficient and its sufficiency is hard to bound using the current theories. Specifically, current approaches are very ***domain–specific*** and as we hypothesized that this might be the case because of $lexical - specificity$ and $structured - specificity$.

Further, the existence of a well defined set of features for a practical task in language processing is a common assumption. It is well known that features extracted from syntactic trees are extremely helpful in semantic role labeling [25]. Current state–of–the–art approaches use millions of features that can be seen as ***carefully planned fit*** of an algorithm with respect to true hypothesis again in terms of $lexical - specificity$ and $structured - specificity$. That kind of an approach, after serious effort in devising a set of features and tuning highly complex discriminative models, ends up in fair performance. But strikingly, performance on the task very fast achieves its ***reasonable upper bound*** and then progress tends to go at very slow pace. For example, if we consider syntactic constituency parsing over twenty years of research most of improvements were in the first couple of years. Afterwards the task hit its upper bound and waited for reasonable improvements twice in a decade. All that in the terms of automatic metrics which are as always controversial. Eventually these approaches failed they ***plausibility test*** (i.e. failed to approximate the solution to a problem to an extent which is practically usable) [3]. Thus we argue that current discriminative approaches for the task of semantic role labeling, as for any other NLP tasks, do not provide any insights about the underlying problem.

Furthermore, mere linguistic resources are very arguable by–themselves. Human driven abstractions should be considered usually as ***incomplete and erroneous representation*** of the underlying linguistic phenomena. Major success

in syntactic parsing [23] exactly gain its high performance and interpretability of some of linguistic phenomena by using that kind of reasoning. Thus approaches that do not take that important factor into consideration have limited scope and ***conceptual upper bound*** in terms of (again) $structured - specificity$.

Our current discussion provides a favorable viewpoint for the supporters of unsupervised learning. If the structures are unrepresentative and inherently hard to model the algorithm that learns them directly from data is a reasonable approach. But then one remembers that we are dealing with the most abstract natural phenomena, in which even the most simple possible tasks can be seen as ***AI–complete*** [11].

On the other hand, successes in fully unsupervised methods are quite ***questionable and hard to interpret***. For example, if you learn constituent–like structures over strings of words should you be evaluated against human–driven structures or in some different form? As it has been shown, the former evaluation criterion is very unfavorable toward unsupervised algorithms even in tasks like POS tagging, where they perform much lower that the supervised approaches. However, if one learns in an ***unsupervised manner*** and then uses the learned structures for some other task, the performance is quite promising. For example, [18] shows that by treating dependency structures as completely unobserved and optimizing them to the task of semantic role labeling one can get results in semantic role labeling comparable to the approach that is using gold–standard dependency structures. Further, [31] provides an example of the good use of semantic roles learned in an unsupervised manner on the task of biomedical question answering. However, even these unsupervised tasks share the assumption that the other levels of linguistic analysis are provided as input to the learning process. Thus the unsupervised learning without any linguistic structures that has a goal to be as predictive for some layer of the linguistic analysis is also doomed to fail. Simply, the ***loss of information***, even though being human incomplete interpretation of the language (i.e. annotated resources), is very much evident. Thus, as we discuss immediately below, we believe that we should use a ***semi–supervised approach***.

5.2 Semantic Role Labeling – semi-supervised approach?

It is clear that purely supervised or purely unsupervised approaches are ***insufficient for modeling linguistic structures***; we will therefore try to devise a semi–supervised approach. In what follows, we examine what is incomplete or obscure (i.e. in terms of provided resources) in semantic role labeling, what should be treated as observed, what incomplete and what unobserved (i.e. in terms of variables defined in the model). Our hypothesized beliefs are driven how by theoretical underpinnings of underlying theories, previous works describing empirical properties also general descriptions about some of the annotated resources (in *English* particularly).

5.2.1 Learn verb classes?

By verb classes, we mean *Levin verb classes*, as *Propbank* annotations are built on them. It is clear that we need a level of ***abstraction among predicates***, from the point of view of dealing with sparsity in natural language and from the point of view that mere semantic decompositions exhibit hierarchical structure. When talking about sparsity, the $1M$ word WSJ of the *Penn* treebank is insufficient in quantity and domain coverage to provide many valuable interpretations. For example, the verb *deceive* occurs only once across all inflectional forms, from which follows that one cannot learn basic alternation patterns from this data alone. However, abstracting away by grouping similar verbs together with respect to some criteria is surely a way to handle this problem. For example, consider alternation patterns of the verbs occurring only one time in the training data from the Figure 5.1. Further, on the same figure for each of the rare verbs we show its synonymous verb that occurs at least in four different $syntactic - semantic\ frames$. We did not take the notion of relatedness from any of the linguistic theories that specify verb classes in one way or the other, but rather just queried a dictionary and picked its most frequent verbal synonym from the training data [1]. As you can see related verbs have much more statistics and thus the hope is that the ***alternation patters*** will be learned even for very infrequent verbs (e.g. *deceive* might inherit alternation patterns from its synonymous verb *mislead*).

One can take the intuition for ***clustering predicates*** from $FrameNet$, where everything is organized into one big hierarchy. Furthermore, from a simple computational perspective, when something is infrequent one should ***smooth*** it using something that is more frequent. Even from a purely linguistic point of view, it is hypothesized that language is exhibiting that kind of an abstraction. Taking into account that $FrameNet$ abstractions are driven by humans and also incomplete and ***not statistically representative***, we ignore the possibility for learning semantic abstractions with them. Further, if we wanted to smooth, statistically speaking we would have to first cluster our predicates with respect

[1] Please see Section 6.2 for the description of the training data.

Syn set	Verb	Linking	Fr
1	***deliberate***	*A0:SBJ*	1
	consider	*A1:OBJ*	46
		A0:SBJ A1:OBJ	44
		A0:SBJ	39
		A1:SBJ	24
		A2:OPRD	21
2	***wrong***	*A1:SBJ*	1
	injure	*A1:SBJ*	7
		A1:PMOD	3
		A0:SBJ	2
		A1:OBJ	1
3	***illuminate***	*A0:SBJ*	1
	light	*A0:SBJ A1:OBJ*	2
		A0:OBJ	1
		A1:OBJ	1
		A1:PMOD	1
		A0:APPO A1:OBJ	1
4	***deceive***	*A0:SBJ*	1
	mislead	*A0:SBJ*	2
		A0:SBJ A1:OBJ	2
		A1:OBJ	1
		A0:OBJ A1:OBJ	1
5	***compliment***	*A0:SBJ*	1
	praise	*A0:SBJ A1:OBJ*	8
		A1:OBJ	1
		A1:SBJ	1
		A0:PMOD	1

Figure 5.1: Linking characteristics of rare verbs that occur only one time in the training data and their verbal synonyms which are more frequent. Here we show only 5 most frequent alternations as there are quite many for some verbs (i.e. *consider*).

to some *discriminative criteria* (e.g. distributional similarity), which is actually a good option but more a quantitative one that a linguistically motivated one. Thus our *learning objective* will try to ***learn verb classes*** motivated by *Linguistic theory of linking*.

5.2.2 Learn linking?

Learning linkings is the main evidence to support the intuition behind *Linking theory* [14]. As it has been shown in many different works [12] there is a ***close correspondence*** between *syntactic* and *semantic* dependencies. For example, consider direct *mapping* from a semantic argument to its direct head via syntactic dependency depicted in Figure 5.2. As you can see there is a *high correlation* in mapping of certain semantic roles and syntactic dependencies (e.g. *A0* is most often *SUBJ*, *TMP* is most often *TMP* and so on..). However, several questions arise: Should the linkings be learned so that they are shared across verb classes?

	A0	A1	TMP	MNR
SBJ	54514	19684	15	7
OBJ	3359	51730	93	54
ADV	162	3506	976	2308
TMP	5	60	15167	22
PMOD	2466	4860	142	62
OPRD	37	5554	1	36
LOC	17	145	43	157
DIR	0	178	15	6
MNR	5	48	13	3312
PRP	9	50	11	6
LGS	2168	36	2	2
PRD	413	830	31	38
NMOD	422	388	25	59
EXT	0	20	2	12
DEP	18	150	25	65
SUB	3	84	4	2
CONJ	198	331	22	8
ROOT	62	147	84	2
	64517	88616	16803	6404

Figure 5.2: Close correspondence of syntactic heads and role labels of predicate arguments [12].

Should we constrain them to be *hard-clustered* or *soft-clustered*? The clear fact is that learning linking alternations across verbs should be de–facto imposed by our learning objective, but we will further aim to this in a form of ***probabilistic reasoning***. The more *evidence* the model gets that some verb should inherit alternations from its corresponding verb classes, the more specified the alternations will be, and vice versa.

5.2.3 Learn cross–class roles?

The *ProbBank* annotation guide clearly disclaims that argument fillers can be seen as ***shared across different predicates*** [22]. However, the hope is the *indication* that the effort was made to do that as stated in [22]. One can easily find pairs of predicates for which some arguments have very similar, if not identical, *syntactic* and *lexical* elaboration. For example, in many *PropBank* sentences *Proto-Agent – A0* is elaborated in the exactly the same syntactic and lexical way. However, some other roles, like the *Proto-Patient – A1* are quite **differently elaborated**. Further, this kind of similarity or dissimilarity is ***exchanged between roles and verbs*** across the whole corpus (i.e. some verb class might have similar roles and some roles might be shared across many verb classes and vice versa). So our hope is that we can incorporate its ***learning*** when it is $beneficial$ and neglect it when it is *misleading*. We depict this similarity in Figure 5.3 for five most frequent verbs in the training data showing their ten most frequent lexical heads for core arguments. For example, verbs *have* and *take* have very similar *Proto-Agent – A0* but not as similar *Proto-Patient – A1*.

5.2.4 Latent roles?

PropBank defines roles that are neither too *general* nor too **coarse–grained**. However, when the arguments become $verb-class-cross-shared$ the ***generality straightforwardly increases***. Further, our argument about insufficiency

A0	A1	A2
	have	
%, bank, companies, company, group, nyse people, president, stocks unit	*damage, effect, impact, loss, problems, profit, sales, shares, time, value*	
	include	
agencies, company, court, honeywell, manufacturers , products, receivers, search, soviets wilson	*charge, down, earnings, gain, gains, operations, provision, sale, shares*	*companies, figures, group, net, number, period, products, quarter, results, stocks*
	make	
banks, bush, companies, company, congress, firm, group, investors, plant, subsidiary	*announcement, bid, decision, decisions, loans, money, offer, payments, products, sense*	*case, entitlement, issue, millionaires, part, president, printer, secret, target, team*
	say	
analyst, analysts, company, dealers, official, officials, people, spokesman, spokeswoman, traders	*agreement, anything, everything, funds, nothing, ondaatje, something, thing, things way*	
	take	
bank, companies, company, exchange, government, investors management, people president, traders	*action, advantage, care, charge, control, position, risk, step, steps, time*	*effect, government, japan, part, place, root, shape, tumble, wellcome, ziyang*

Figure 5.3: Ten most frequent lower-cased surface forms for five most frequent verbs in the training data.

of the human–driven abstraction, as applied to $verb-classes$, applies here as well. The ***level of generality*** of semantic roles is the subject of an ongoing debate in the field of *lingustics* and will most likely to remain as such. As we saw in Section 2, two widely accepted standards are $PropBank$ and $FrameNet$. We argue that one should ***directly reason*** over the ***level of granularity*** of semantic roles as represented and constrained by the *linkings* and $verb-classes$. We see the level of granularity both as $domain-specific$ as the $Airport$ example [2] and as general as the two $Proto$ roles [2] to be undoubtedly justified and representative, as long as it is constrained by the overall model with having the highest likelihood. That kind of reasoning drives the semantics to be as *self-expressive* as possible. Further, in Figure 5.5 we show ten most frequent lexical role fillers for core arguments across all predicates. The clear fact which is immediately evident is that as for some role fillers (e.g. $A0$ or $A1$) the *semantic relatedness* is present, while for some other (e.g. $A3$ or $A4$) it is quite questionable.

5.2.5 Learn syntactic classes?

Syntactic classes are arguably helpful because the mere connection between *syntax* and *semantics* is a *latent* one (i.e. hard to specify). In some cases the two map almost *deterministically*, but in some other drive $falsifiable\ clues$. We take *dependency path* from an argument to the predicate as our syntactic repre-

[2] See Section 2.1.

sentation of a predicate argument. This dependency path is gained by *traversing* the dependency tree from the predicate to the argument while concatenating all dependency relations in between them and marking them with 1 if the node is not the argument in question and 0 if it is. [3]. ***Learning the latency*** by introducing latent variable between semantics or linking variable and syntax in our model we will try tackle ***the learning of the linking via syntactic class***. Specifically, we assume these dependency path to be a reflection of the underlying *syntactic function* of semantic arguments. Thus the model should learn what does it mean in this particular *syntactic theory* to be a *subject* or an *object*. Even though that might not be *syntactically sound* it should only say what is the usual representation for the particular semantic role. Further, in cases when we observe preposition as a candidate argument we take its immediate right most child as the lexical filler and *enrich* the dependency path by concatenating it with the observed preposition [3]. We take this motivation from unsupervised approaches which work on the principle of keys that can be simply viewed as enriched syntactic dependencies (i.e. with aspect, preposition). Some graphical motivation to the above discussion is depicted in Figure 6.2 where we show top ten dependency paths between a core argument and a predicate across all verbs.

A0		A1		A2	
SBJ0	19305	*OBJ0*	31627	*OBJ0*	1583
VC1SBJ0	6617	*SBJ0*	5585	*EXT0*	1161
IM1OPRD1SBJ0	1101	*VC1SBJ0*	4934	*OPRD0*	541
NMOD1PMOD1PMOD0	984	*APPO1PMOD1PMOD0*	1578	*SBJ0*	457
IM1OPRD1VC1SBJ0	847	*NMOD1PMOD1PMOD0*	1548	*VC1SBJ0*	264
CONJ1COORD1SBJ0	745	*VC1VC1SBJ0*	1473	*NMOD1PMOD1PMOD0*	228
IM1OPRD1OBJ0	695	*NMOD1OBJ1OBJ0*	781	*PRD0*	197
NMOD1SBJ1SBJ0	690	*APPO1SBJ1SBJ0*	650	*NMOD1OBJ1OBJ0*	135
ADV1SBJ0	642	*APPO1OBJ1OBJ0*	569	*PMOD0*	129
NMOD1OBJ1OBJ0	419	*PMOD0*	566	*NMOD1SBJ1SBJ0*	111

Figure 5.4: Top ten dependency path between a core argument and a predicate acrross all verbs.

5.2.6 Learn word classes?

To model the ***appropriate level of granularity*** between *semantic roles* and thier lexical representations, one certainly needs some form of $class - based$ definition. First of all, the ***lexical sparseness*** is a ubiquitous problem; $Zipf's$ law holds in all languages and is one of the main problems in language processing. Abstracting to a level of granularity of grammatical categories or any other

[3]For an example see Section 5.4.1.

A0		A1		A2		A3		A4	
company	1302	%	621	%	975	money	14	home	9
companies	501	shares	568	president	98	chairman	11	wells	6
officials	483	prices	510	points	87	company	9	victim	3
investors	478	sales	500	cents	72	advertisers	5	way	3
people	453	market	410	results	55	director	5	place	2
analysts	373	company	326	chairman	35	dollars	5	burns	1
government	359	stock	314	place	35	injunction	4	number	1
group	330	rates	301	point	35	pounds	4	periods	1
spokesman	317	bonds	292	group	32	taxpayers	4	something	1
traders	317	profit	276	shares	28	today	4	toddler	1

Figure 5.5: Ten most frequent lower–cased surface forms for all verbs in the training data.

stochastically derived form is an option toward handling the problem of sparsity. However, we will incorporate ***learning of lexical classes*** in the model in the space of surface form driven latent variables between the *semantic* and *lexical* information. Such an approach can be as well motivated by the intuition that language should exhibit a form of semantics on the level between the $surface\ form$ and $frame\ semantics$ (e.g. in some sense present in $Framenet$), clearly consistent with our ***philosophy of hierarchical representations of semantics*** on varying levels [4].

5.2.7 Learn word senses?

$Word\ senses$ as provided by the $PropBank$ are, to a great extent, already ***modeled in our approach***. It is rather clear that trying to differentiate between $numbers$ of semantic arguments of a predicate will not change anything, since such are already modeled by the learned linkings. Further, the overall compactness of the model will ***constrain the allowable syntactic and lexical specifications*** in such a way that they will capture much of the sense meaning. Furthermore, specifying senses for $infrequent\ words$ is a huge problem and ***learning word sense implicitly*** in the model of our grounding is reasonable approximation.

[4] In the terms of $Framenet$ the model should learn appropriate representation for units.

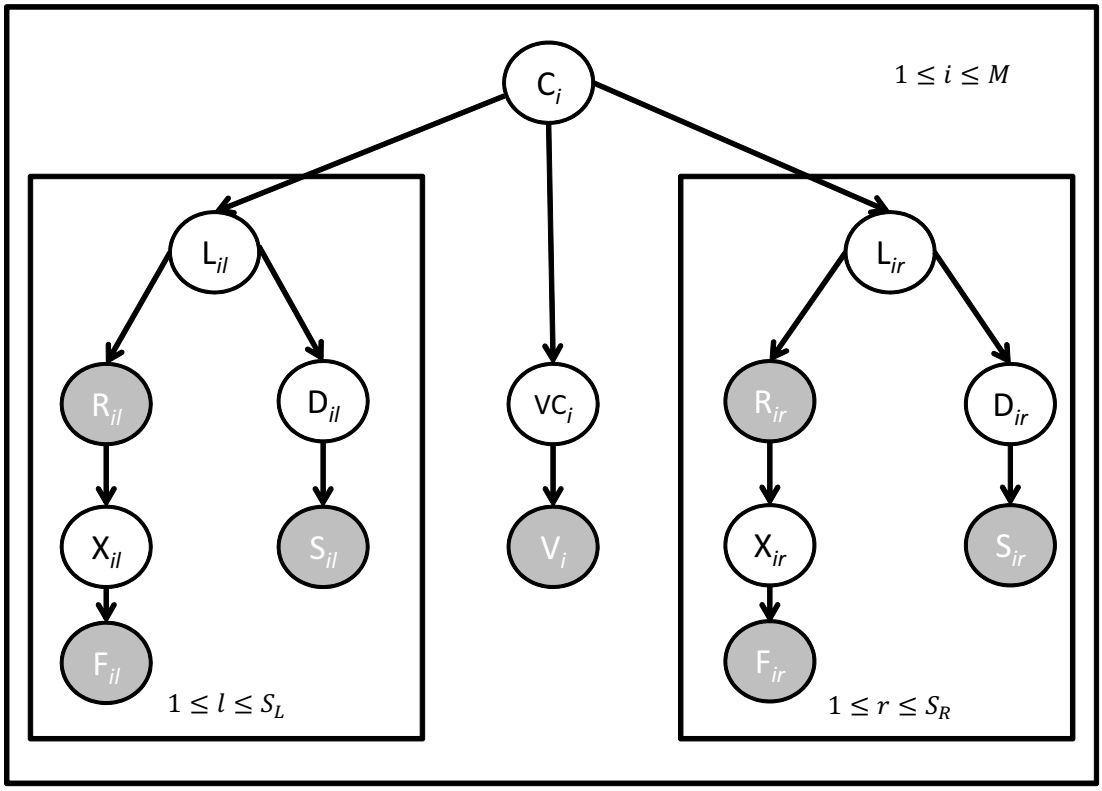


Figure 5.6: Linguistically motivated graphical model for semantic parsing: Model 2. Note that we do follow usual graphical model representation. However, as the model is very general in fact most of the variables could be in some cases observed/unobserved depending on the motivation behind the usage. Where: M – is the number of propositions, S_L and S_R – are numbers of left and right arguments respectively for a particular proposition.

5.3 Models

In order to accomplish our ultimate goal of ***learning latent information between many layers of linguistic knowledge***, we argue about the modeling perspective in the domain of ***probabilistic models***. Our main goal is to learn *varying abstractions* of *semantics* and their corresponding *coupling* with *syntactic* and *lexical* information. Further, our modeling problem has two folds: in one it tries to learn *abstractions* and *generalizations* about verb classes, linking, role fillers and word classes and in the second *specifications* and *encapsulations* of elaboration of semantics in its lexical and syntactic form. That makes defining a model and its corresponding learning objective quite difficult.

In what follows, we argue that the appropriate level of abstraction and encapsulation can be found in ***close correspondence of semantic and syntactic dependencies***. As it has been shown by [12][13], there is a very high correlation of mapping between syntactic and semantic dependencies. Furthermore, in their approach, which is formulated as an unsupervised learning problem, lexical information plays a crucial role in the unsupervised discovery of semantic role fillers. An unsupervised approach presented in Section 3.2 is also using a minimal level of interaction between syntax and semantics. A further recent success in unsupervised semantic parsing is achieved by $cross--verb\ clustering$, where the

syntactic dependencies are the main information used [24] [30]. We thus argue that one does not need to fully specify the complete lexical and syntactic derivation of the semantic elaboration, and the most important aspects can be found in ***minimum correspondence***. We devise two *graphical models* that represent different beliefs about appropriate structuring of the domain of interest.

A model that represents ***linguistically plausible*** interaction of syntactic and semantic dependencies is depicted in Figure 5.6. First, one generates a frame class – C, its corresponding linkings and the verb class – VC. A linking is directly represented as in *Linguistic theory of linking* by interaction of semantics – R and syntax – D. Here semantics is represented by a semantic role which is further specified by the its lexical class – X. Lexical class in this case can been seen as a *selection preference* property of a semantic role and its directly generating surface form of an argument – L. Further, syntactic variable D is in fact a syntactic class variable which should incorporate intuition that the close correspondence between semantics and syntax is not possible to be *trivially specified*. Thus this class variable groups similar linking properties of dependency links – F between a semantic role and its predicate. Hopefully, this variable will learn what does it means to be a subject or a direct object when ones specifies a linking for some of the semantic roles. Further, verb class variable – VC generates its verb – V. Thus should corresponds to the intuition that verbs occurring in similar frames should group together and that on the other hand same verbs can occur in different frames.

Note that the linking is not fully specified by the frame class variable in this graphical model as the order does not influence generation of the linkings. It is only probabilistically constraining the frame in at least the number and the type of arguments. Further, verb class variable and frame class variables are adapted more for *explanation purposes*, as they are in fact two very closely related variables and they specify ***complementary information***. If we would like to be as ambitious we would say that the frame class should be predictive of the frame, as in the sense of $FrameNet$ at some level of abstraction, while the verb class variable would then in fact specify an event on some level of abstraction.

Our other model, which represents more ***computationally plausible*** interaction of syntactic and semantic dependencies with the goal of being predictive of the semantics is depicted in Figure 5.7. The main difference is that the frame class – C instead of generating linkings generates semantic roles R directly from the frame. Thus linking alternations are taken to be as a partial explanatory derivative for the semantics rather that main source of $syntactic - semantic$ interaction. We thus ***reduce*** some ***uncertainty about semantic representation*** as it is more constrained by the overall model structure. Other variables have the same meaning and take the same set of values.

We depict two ***generative stories*** in more detail in the Figure 5.8 [5]. All distributions specified are multinomial ϕ directly defined on the dataset F.[6] Thus

[5]For an example of derivation see Section 5.4.1.

[6]Check appendix A for more detailed info about the notation used in generative models and grammar formulations.

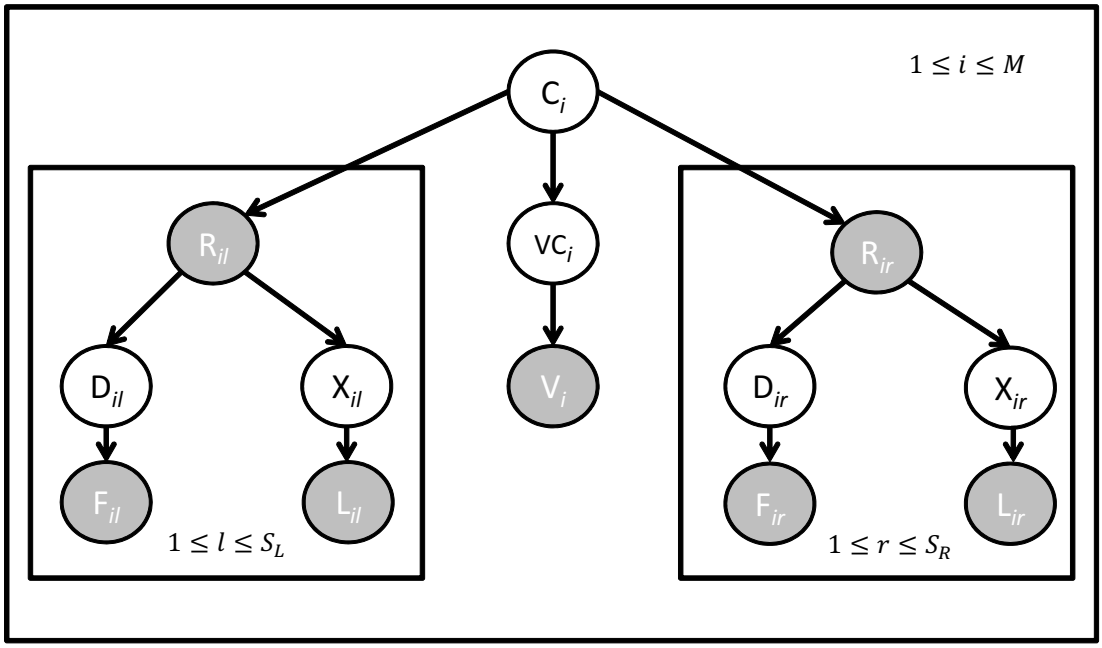


Figure 5.7: Computationally motivated graphical model for semantic parsing: Model 1. Note that we do follow usual graphical model representation. However, as the model is very general in fact most of the variable could be in some cases observed/unobserved depending on the motivation behind the usage.

the instantly noticeable difference with usual models for semantic role labeling is that we ignore *frame/predicate* specific distributions. Further, as we do not model argument identification we never have any *uncertainty* about the number of arguments and they are directly defined by sampling the frame. The implausibility of ignoring the order of linkings in also depicted as we do not have any variables modeling that. Also, note that we did not specify the unobservability of the frame, verb class and other variables they can be trivially excluded from the story as they do not impose any uncertainty. But, with motivation of *generality* we have fully specified them as well. Generative stories are completely the same with respect to the corresponding graphical model: $Model$ 1 – First the frame class is sampled – f. Further on the choice of the frame we sample the verb class – vc and for each of arguments in the frame – $Args(f)$ we sample a role – r. Conditioned on the role syntactic – d and lexical – x classes are sampled. Ending up in sapling the surface form of an argument – l, dependency link – s and the verb – v. $Model$ 2 – Only difference with respect to Model 1 is that frame defines linkings – l which conditions sampling of the role – r and the syntactic class – d.

As our model is currently specified, it is a very simple model indeed. One could easily argue that it cannot capture many phenomena in natural language that are influenced by the same type of linguistic knowledge. However our goal is not to fully specify all the forms of linguistic knowledge that we are using but rather only one: ***Semantics*** (i.e. semantic roles).

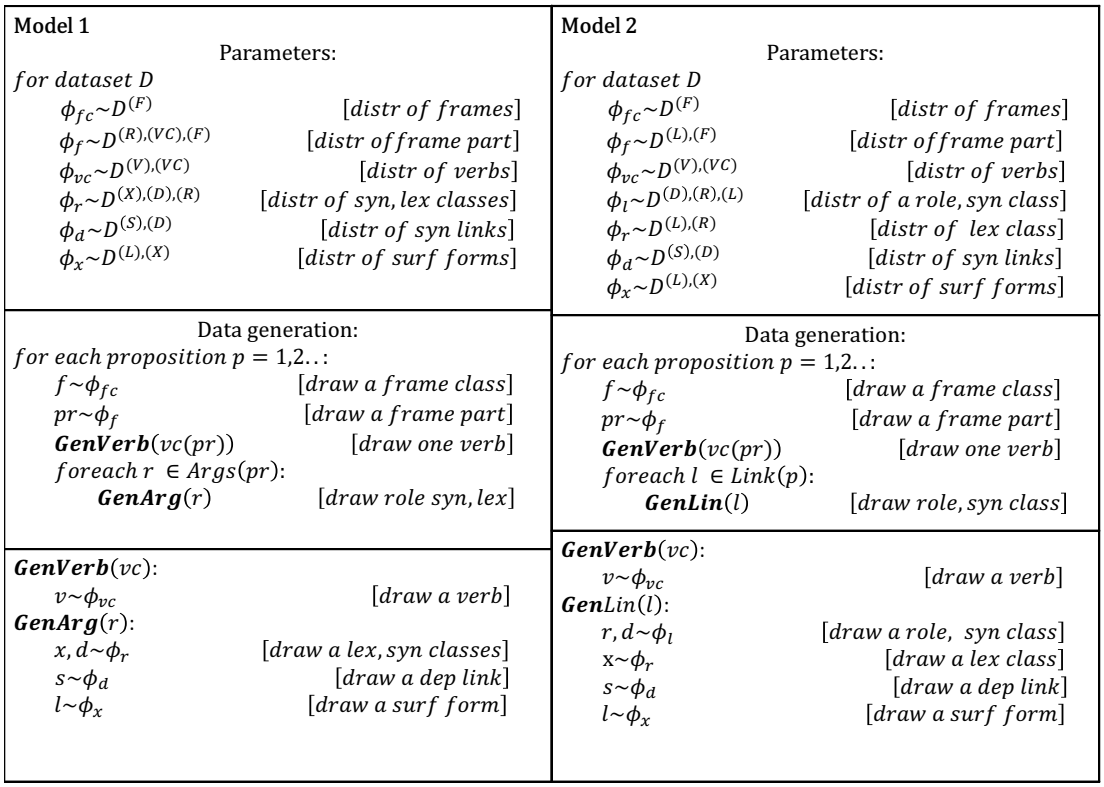


Figure 5.8: Generative models for semantic parsing.

5.4 From the modeling to reality

As we are after semantics model should be only ***predictive for semantic roles*** given all other observable arguments. Further, even in this very simple model we have three *unobservable* types of variables. We do not get to observe $frame-class$, $verb-class$ and $word-class$ variables. In the simplistic type of models as the ones on Figures 5.6 and 5.7, that would limit the ***expected performance*** of the model in a high degree. Consequently, we tackle the described problems as well as the full motivation behind the modeling by ***adapting latent variables on each cluster node*** (i.e. cluster nodes are all nodes who have children). Then the $non-observability$ of the variables in our model actually becomes its ***expressive power***. Further, all of our cluster variables are shared across different semantic frame instances, on the sentence level as well as on the corpus level. Thus the model will be able to ***learn varying degrees of semantic knowledge*** as represented by all: $frame-classes$, $verb-classes$, $syntactic-classes$, $word-classes$ and $linkings$. Also note that now, since we do not observe the cluster variables, their children variables lose their *independence assumptions*. Further, from a probabilistic point of view, the model is *very compact* thus the correlation between variables should be stronger and its learning easier. Most importantly, our model does not use any features so the model is $language-independent$; no changes in the model are required to handle new language instances.

Our graphical model could be *realized* in many different forms of ***probabilistic learning and inference***. First let us consider what the current

Grammar Model 1:		Grammar Model 2:	
$CFG\ Model\ 1$		$CFG\ Model\ 2$	
$S \rightarrow F$	$[generate\ frame]$	$S \rightarrow F$	$[generate\ frame]$
$F \rightarrow RL\ \ VC\ \ RR$	$[generate\ frame\ part]$	$F \rightarrow AL\ \ VC\ \ AR$	$[generate\ frame\ part]$
$VC \rightarrow V$	$[generate\ verb]$	$VC \rightarrow V$	$[generate\ verb]$
$RL \rightarrow RL\ \ R$	$[generate\ roles\ left]$	$AL \rightarrow AL\ \ L$	$[generate\ linkings\ left]$
$RL \rightarrow e$		$AR \rightarrow AR\ \ L$	$[generate\ linkings\ right]$
$RR \rightarrow RR\ \ R$	$[generate\ roles\ right]$	$L \rightarrow L\ \ R\ \ D$	$[generate\ role, syn\ class]$
$RR \rightarrow e$		$L \rightarrow e$	
$R \rightarrow X\ \ D$	$[generate\ syn, lex\ classes]$	$R \rightarrow X$	$[generate\ lex\ class]$
$X \rightarrow L$	$[generate\ dep\ link]$	$X \rightarrow L$	$[generate\ dep\ link]$
$D \rightarrow S$	$[generate\ surf\ form]$	$D \rightarrow S$	$[generate\ surf\ form]$
$where$:		$where$:	
$R \in VOC(roles),\ L \in VOC(role\ forms)$		$R \in VOC(roles),\ L \in VOC(role\ forms)$	
$S \in VOC(dep\ links),\ V \in VOC(verbs)$		$S \in VOC(dep\ links),\ V \in VOC(verbs)$	
$X \in \mathbb{N}, e - empty\ set$		$X \in \mathbb{N}, e - empty\ set$	

Figure 5.9: Contex free grammars of Models 1 and 2.

model cannot capture. As it is represented the model does not incorporate any *prior knowledge* on any type of variables. For sure that kind of information is very useful in reasoning over *linguistic structures*. Many variables, if not all, from our model could valuably incorporate *priors*. However, one of them has been shown very crucial in dealing with semantic role labeling. The *linking prior*, which is the main component of the unsupervised model from Section 3.2, has been used in semantic parsing since its pioneered work in the task [7] till the state–of–the–art models in unsupervised semantic role labeling of the recent research [32]. Further even though our model is compact in dealing with semantic role labeling it cannot guarantee or assume the *number of hidden variables*. That implies that some form of ***split-merge adaptation*** that can *automatically* adapt at the training data should be used. Further as for all latent variables models the *intractability* imposed the need for well tackled and implemented approximate inference algorithms. Finally, we decide to use ***Latent Probabilistic Context–free Grammar*** – $PCFG-LA$ as the formalism for the *model realization*. With observing our model as a $PCFG-LA$ we naturally *capture* priors over structures as the they are defined over $context-free$ rules and in that way we tackle learning of linking compactly in our formulation. Further, $PCFG-LA$ have been so far very successfully used in problems of syntactic parsing and have developed advanced learning and inference procedures. One of them is developed in the *Berkley parser* implementation of $PCFG-LA$. Thus we only need to formulate our problem in terms of parsing with Context-free Grammar and we use the *Berkeley parser* [7] as the of–the–shelf tool.

Grammars corresponding to $Model\ 1$ and $Model\ 2$ are depicted in the Figure 5.9. The main difference to be noted is that the models in the form of a Context–free Grammar do not have the same $form$ as the models depicted in the generative story on the Figure 5.8. Specifically, this difference is in generating roles – R or linkings – L and verb classes – VC from the frame class – C. Also the difference is in generating the syntactic – D and lexical – X classes or roles

[7] http://berkeleyparser.googlecode.com/files/BerkeleyParser.jar

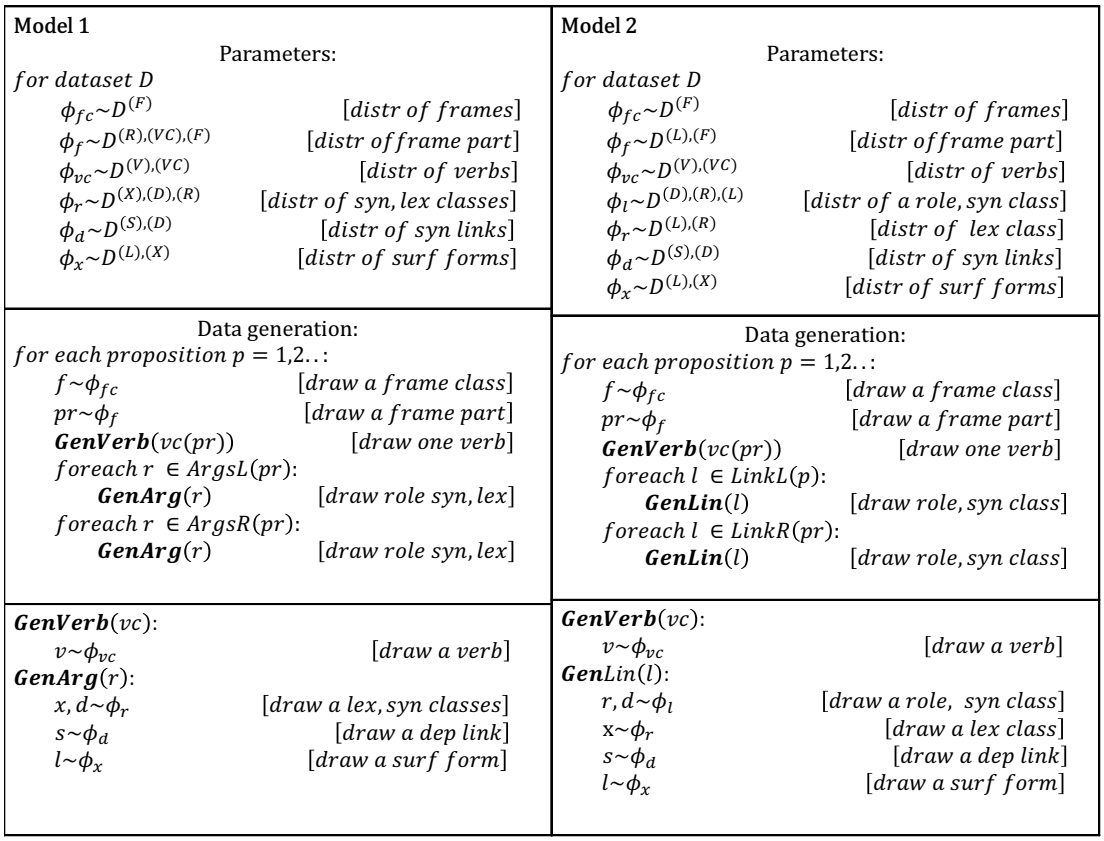


Figure 5.10: Generative stories of models altered by conversion to Context–free Grammar.

– R depending on the model. The difference is that the order is no longer irrelevant and that the variables are ***jointly generated*** from corresponding parents. Consequently, that provides additional expressive power in capturing the linking alternations and further even more closely coupling the syntax and semantics on lower levels.

The generative stories of these ***altered models*** are shown in Figure 5.10. Main difference, as already noted, is that frame – f generates ordered list of participants – pr: linkings – l or roles – r; with respect to the position of a verb class – vc : $ArgsL(f)$ - left, or $ArgsR(f)$ - right. Also, for the $Model$ 1 lexical – x and syntactic – d classes are sampled together from the governing role – r.

As we use latent variable $PCFG$ (i.e. $PCFG{-}LA$) the grammar form changes according to the ***latency***. Thus, the new form of models in $PCFG - LA$ form is depicted in the Figure 5.11. Main difference, of course, are unspecified non–terminal symbols with an integer value – $[X]$.

The ***formal form*** of the models follows $PCFG - LA$, as we observe each semantic frame as a tree in the Context-free Grammar form. Thus the mathematical underpinnings are already defined and the reader is encouraged to see Section 3 for related references. One can easily see many alternations of our model as depicted by some linguistic property. For example, we could instead of word class – X observe some word class from an external clustering (i.e. $Brown$ classes). The verb could be represented by its lemma or surface form as well as the arguments'

Latent Grammar Model 1:	Latent Grammar Model 2:
$CFG - LA\ Model\ 1$	$CFG - LA\ Model\ 2$
$S \rightarrow F[X]$ $[generate\ frame]$	$S \rightarrow F[X]$ $[generate\ frame]$
$F[X] \rightarrow RL\ \ VC[X]\ RR$ $[generate\ frame\ part]$	$F[X] \rightarrow AL\ \ VC[X]\ AR$ $[generate\ frame\ part]$
$VC[X] \rightarrow V$ $[generate\ verb]$	$VC[X] \rightarrow V$ $[generate\ verb]$
$RL \rightarrow RL\ \ R[X]$ $[generate\ roles\ left]$	$AR \rightarrow AR\ \ L[X]$ $[generate\ linking\ right]$
$RL \rightarrow e$	$AL \rightarrow AL\ \ L[X]$ $[generate\ linking\ left]$
$RR \rightarrow RR\ \ R[X]$ $[generate\ roles\ right]$	$L[X] \rightarrow R[X]\ \ D[X]$ $[generate\ roles, syn\ class]$
$RR \rightarrow e$	$L[X] \rightarrow e$
$R[X] \rightarrow X[X]\ \ D[X]$ $[generate\ syn, lex\ classes]$	$R[X] \rightarrow X[X]$ $[generate\ lex\ class]$
$X[X] \rightarrow L$ $[generate\ dep\ link]$	$X[X] \rightarrow L$ $[generate\ dep\ link]$
$D[X] \rightarrow S$ $[generate\ surf\ form]$	$D[X] \rightarrow S$ $[generate\ surf\ form]$
where:	*where*:
$R \in VOC(roles)$, $L \in VOC(role\ forms)$	$R \in VOC(roles)$, $L \in VOC(role\ forms)$
$S \in VOC(dep\ links)$, $V \in VOC(verbs)$	$S \in VOC(dep\ links)$, $V \in VOC(verbs)$
$X \in \mathbb{N}$, e – *empty set*	$X \in \mathbb{N}$, e – *empty set*

Figure 5.11: Latent Context–free Grammar forms of models. Note that we omit latent annotation symbols on proxy variables RL, RR, AL and AR from the parent frame class variable F as the latency is actually induced on F using these symbols to tackle varying number of arguments.

lexical heads. Further, one could incorporate POS tag information on arguments as well as on predicates. Finally, coupling unsupervised variables from our model with linguistic resources (e.g. $FrameNet$, $VerbNet$) would result in increased model word–awareness and hopefully boost performance.

5.4.1 Conversion process

The training data consist of a set of sentences annotated with *pos tags*, *syntactic dependencies* and *semantic roles*. An example sentence we show in Figure 5.12. As we are using $PCFG-LA$ grammar for semantic parsing we need so specify how one gets a constituency tree for semantic parsing from the training data. The process is very simple and intuitive as we starting from graphical models from figures 5.6 and 5.7 just need to specify *observable* variables. Thus variables: F, L, V and R.

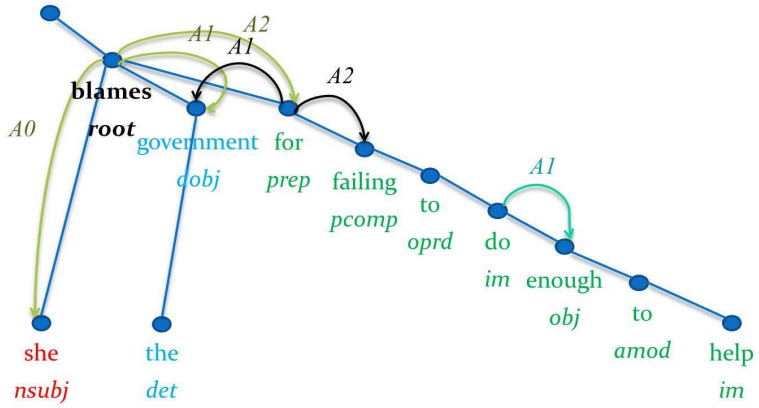


Figure 5.12: Sample sentence from training data annotated with syntactic and semantic dependencies.

As you can see in the Figure 5.12 we have a verb *blame* and its two arguments. Simply we strip of from the annotated tree required information by traversing the tree and filling the corresponding constituency tree. Thus we first pick the first argument – $A0$ and specify its lexical filler – L with the assignment *she* (we use lowercased surface forms for the lexical fillers) and syntactic class – F with assignment $nsubj$. Further, we specify the verb – V with assignment *blame* (verbs are always represented with the predicted lemmas in our model). Similarly as we did for $A0$ we specify $A1$ by its lexical filler – L with assignment *government* and syntactic class D with assignment $dobj$. Finally, we show the only special case when we observe preposition as the lexical filler of the semantic argument. In this case it is the preposition for in the role of $A2$. Correspondingly, we specify the lexical filler – L with the right most child – $failing$ in this case and syntactic class – D by additionally concatenating the dependency path – resulting in the example with an assignment $prep-for$. You can see the ***converted constituency tree*** ready for semantic parsing in the Figure 5.13 by the graphical form of the $Model$ 1.

We obtain the constituency tree in the form of the $Model$ 2 by just using different skeleton structure of the constituency tree. Specifically, the observed variables in both of the models are the same. The constituency tree for semantic parsing by $Model$ 2 is depicted in Figure 5.14.

In the ***inference phrase*** we observe terminals and preterminals from these trees. Thus we just run the *Berkley parser* on known POS tags.

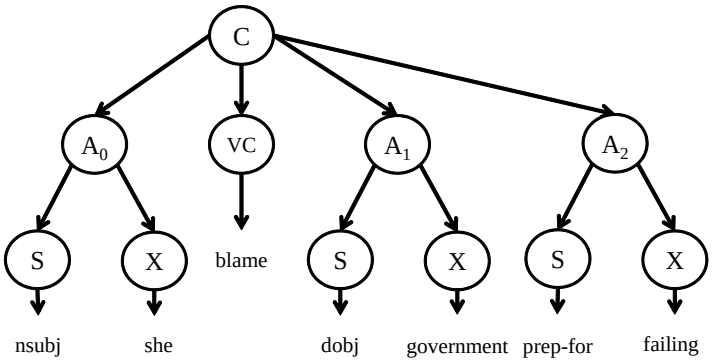


Figure 5.13: Constituency tree for semantic parsing by Model 1. This corresponds to the model before learning – thus before we have learned latent annotation symbols for each non–terminal.

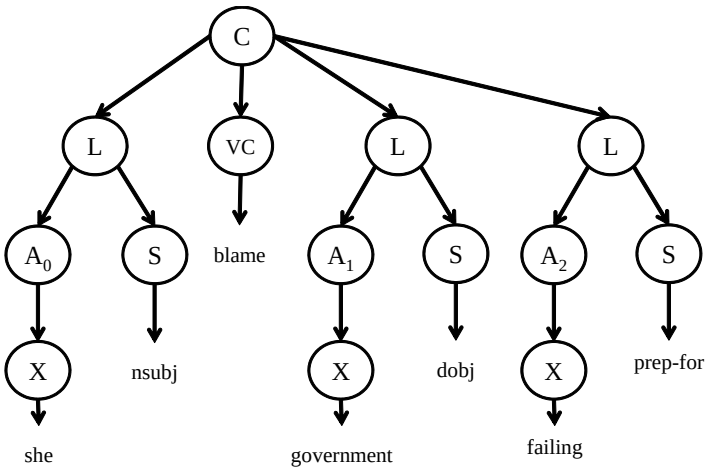


Figure 5.14: Constituency tree for semantic parsing by Model 2. This corresponds to the model before learning – thus before we have learned latent annotation symbols for each non–terminal.

6. Empirical plausibility

6.1 What has been learned

In order to ***qualitatively examine*** our model in search for some clues of *linguistic*, *semantic* or any other nature we run our model on training data and keep the final (i.e. most probable) clustering of variables with respect to the inference. That means that we cannot possibly show what has been fully learned in the model but only partially as we do not assume ***exclusive clustering*** of any of information learned by our model. Further, even this *partial clustering* (e.g. semantic roles across multiple clusters) which in its essence is of probabilistic nature is influenced by all other clusterings in the model. Thus cutting the probabilistic nature of our model is constrained by the means of evaluation and in fact we would assume that for different tasks, possibly different objectives and different human derivation the model will derive the different meaning of induced structures. Whilst the *model* – the *grammar* – is abstractly probabilistic till its appliance. And as we do not optimize our model to predict semantic roles directly we would assume clusters to hide a lot of valuable information which could be used across many ***different tasks***.

As our model arguably tries to learn *hierarchical semantic information* as represented by the *lexical* and *syntactic* elaboration we argue that the resource which would be directly appropriate for the evaluation of such approach does not exists and cannot exists. Reasons being that devising that kind of hierarchical representation would be impossible for humans as its real nature is in close correspondence of syntactic and lexical information as derived by the distributional characteristics of the language phenomena which is far from being understood (without even thinking about cross–linguality). It is clear that the $verb-per-verb$ basis falls back in the first thought and we decide to use the closest corresponded ***FrameNet***. As one might argue that *Propbank* or $VerbNet$ might be more suitable for the direct comparison because of the clearer annotation setting our assumption is that $FrameNet$ has closer overall correspondence with the abstract notions which we are trying to devise. Thus we will try to qualitative compare model assignments on various levels with the $FrameNet$ annotation [1].

When we run our model on the training data we can observe which cluster has been used as the representation of the predicted proposition. Thus a single cluster (e.g. $A0-10$: $A0$ role with assignment of cluster 10) has different neighboring variables associated (e.g. $A0-10$ might be in frame $C-5$, with elaboration on lexical class $X-3$ and syntactic class $S-55$, which further could take values: *president* and $SBJ0$ respectively) and further different prepositions for a single cluster can take same values (e.g. *John* could be found both in $A0$ and $A1$). Consequently, to get some insights into the ***representational property*** of clusters we consider only some number of the most frequent assignments (10 is the usual number) with respect to some of the neighboring variables (e.g. we might consider what are 10 most frequent lexical assignments to the cluster $A1-10$).

[1] Note that we are learning on *PropBank* annotations and evaluating on $FrameNet$.

Lexical classes
X-27
bond, earnings, income, index, price, profit, revenue, sales, share, stock
X-38
ago, for , friday, in, month, over, today, week, year, yesterday
X-1
analyst, executive, investor, manager, official, president, spokesman, spokeswoman, those, trader

Figure 6.1: Top ten assignment to a some of the lexical classes clusters with respect to their surface representation.

We begin our examination with ***lexical classes X*** with respect to their elaboration via surface form L depicted in Figure 6.1. Now as for all of our examination we take the surface represented in the cluster and query the $FrameNet$ [2]. Unfortunately, we fail already on our first trial with the cluster $X-27$ and its surface form *bond*. As the *bond* is defined in $Bail\_decision$, $Social\_connection$, $Connectors$ and $Attaching$ lexical units non of which has to do with its financial sense. However, our query also returns one frame $Bond\_maturation$ which is carrying required economical sense but even though in the example of the frame (which deals with time and legal property of the guarantee of the transaction) we can find its usage it has not been specified as the unit element[3]. For our next surface form *earnings* we have more luck and retrieve its unit in the frame $Earnings\_and\_losses$. There we can find other lexical units defined by the frame and try to match them with the content of our cluster. And indeed find a match with $income$, $profit$ and $revenue$. Further, while $index$ and $sales$ are undefined as units in $FrameNet$, we find $share$ and $stock$ as the part of $Capital\_stock$ frame. Note now the following claim which will hold throughout the discussion: *When we constrain our cluster's representation via structural assignment we directly* ***influence its meaning and its interpretability***. Thus in order to get correspondence with the $FrameNet$ annotations our required structured conditioning would have to take all of variables in our model (possibly excluding syntax S). And as this claim holds (as we will show it a bit later) the close similarity between our lexical classes cluster top assignments and the $FrameNet$ indicates that already on the very low level of abstraction we have fairly good annotations.[4] The same behavior can be find for other two example clusters $X-38$ and $X-1$.

Next consider ***syntactic classes D*** and their dependency path representation F shown in Figure 6.2. In this case we cannot argue while comparing with $FrameNet$ nor even $PropBak$ thus we take a different path and examine the cluster meanings with respect to its connection to the semantics R – the *linking* (i.e. semantic roles). We find out the cluster $Y-19$ is the argument which is on the left from the verb and is usually represented as the $A1$ – thus arguably

[2]Try it by yourself: `https://framenet.icsi.berkeley.edu/fndrupal/`.

[3]Please note the incompleteness of $FrameNet$ and the gains of possible automatic induction driven with current resources!

[4]Note that we have selected all of our clusters randomly and that the infrequent forms got suppressed in the assignments of clusters.

Syntactic classes
Y-100
AMOD0, LOC0beyond, NMOD1AMOD1PRD1PRD0to, OBJ0that, OBJ0to, OBJ0whether, OPRD0to, PRP0to, SUB0, VC1PRP0to
Y-19
APPO1OBJ1OBJ0, APPO1PMOD1PMOD0, APPO1PRD1PRD0, APPO1SBJ1SBJ0, IM1NMOD1OBJ1OBJ0, IM1NMOD1PMOD1PMOD0, IM1OPRD1APPO1OBJ1OBJ0, IM1OPRD1APPO1PMOD1PMOD0, IM1PRD1SBJ0, IM1PRP1OBJ0
Y-22
CONJ1COORD1CONJ1COORD1SBJ0, CONJ1COORD1COORD1SBJ0, CONJ1COORD1SBJ0, COORD1COORD1SBJ0, COORD1SBJ0, IM1OPRD1SBJ0, OPRD1OBJ0, PMOD1TMP1SBJ0, SBJ0, VC1SBJ0
Y-15
AMOD1PRP0because, CONJ1COORD1PRP0because, IM1OPRD1PRP0as, PRP0as PRP0because, PRP0since, TMP0because, VC1DEP0, VC1PRP0because, VC1TMP0before
Y-99
ADV0at, ADV0for, ADV0from, ADV0on, ADV0with, CONJ1COORD1OBJ0, COORD1OBJ0, GAP-OBJ0, LOC0in, OBJ0

Figure 6.2: Top ten assignment to a some of the syntactic classes clusters with respect to their dependency path representation.

Proto–Patient. Further, $Y-22$ is mostly representing the Proto–Agent $A0$ irrespectively with respect to the position to the verb. While the cluster $Y-99$ mostly represents Proto–Agent $A0$ on the right from the verb. Finally, $Y-15$ exclusively denotes adjunct argument $AMCAU$ – cause adjunct.

Further, let us consider distribution of ***semantic classes R*** with respect to the surface realization as depicted in Figure 6.3. It is clearly obvious that $A0-35$ contains some of the entities involved in large financial transactions (as we could speculate with respect to domain of $PropBank$). Then $A0-4$ contains some of the $Leadership$ frame units in terms of $FrameNet$: $president$ and $official$; and some from the frame $People\_by\_vocation$: $manager$ and $trade$.

Semantic role classes
A0-35
britain, germany, goldman, lynch, malcolm
A0-4
company, democrat, investor, man, manager, official, people, president, those, trader
A1-17
income, market, price, rate, rates, revenue, sales, share, stock, stocks
A1-21
arney, brawl, goodrich, intensity, lesk, novell, rowland, shah, tonnage, wedtech
A2-4
%, consistently, rapidly, sharply, slightly, slowly, so, steadily, widely, with
A2-19
clear, difficult, easy, impossible, intact, necessary, possible, tough, unclear, unlikely
A3-25
banks, buying, company, germany, holding, operations, profit, revenue, stake, time

Figure 6.3: Top ten assignment to a some of the semantic classes (i.e. semantic roles) clusters with respect to thier indirect surface realization while summing over lexical classes.

Verb classes
V-28
believe, decide, fear, hope, know, mean, see, show, suggest, think
V-10
acquire, buy, earn, have, hold, make, post, report, sell, take
V-6
boost, carry, cause, mark, post, produce, provide, reach, report, yield

Figure 6.4: Top ten assignment to a some of the verb classes clusters with respect to their lemma realization.

While both are at least intuitively part of the frame *people* [5] which, of course, contains units *people* and thus *they*. Further, as the *democrat* [6] is not defined as a unit we can only speculate its connections to the *Leadership* and *People_by_vocation* frames as being intuitively connected. Finally and mysteriously *company* is disconnected from both of these frames and its a part of stand–alone *Commerce_scenario* which is a separate domain in *FrameNet*.

Cluster $A1 - 21$ contains Named entities; $A1 - 17$ units of commerce–like units from which most are *Goods* and $A2 - 4$ adverbs representing some forms of rate or speed. Units of three inheritance frames of the frame *Gradable_attributes* are dominating in the cluster $A2 - 19$. From which *tough*, $difficult$, *easy* and *impossible* are from $Difficulty$ frame; *impossible*, *possible* and *unlikely* are from the frame *Likelihood*; and *clear* and *unclear* are from *Obviousness* frame.

We continue our examination by considering distribution of ***verb classes VC*** as represented by verb lemmas V (See Figure 6.4). Starting with cluster $V - 28$ we see that the verbs $feel$, *believe*, and *think* can be found in the frame *Opinion*, verbs like *know*, *believe* and *think* are in the frame *Awareness* and thus interleaved. Further, in the frame *Evidence* we find verbs *mean* and *see* and concluded that the cluster is not very much pure as represented with verbs alone and that all verbs are indeed very abstract. In the cluster $V10$ we observe instance from two general frames *Giving* and *Getting*.

Frame-verb classes
CV-56
believe, know, mean, report, say, see, show, suggest, tell, think
CV-47
buy, get, have, include, make, own, reflect, sell, take, yield
CV-40
act, close, come, do, go, move, run, sell, walk, work
CV-44
climb, come, decline, drop, fell, go, grow, increase, jump, rise

Figure 6.5: Top ten assignment to a some of the frame classes with respect to their verb lemma realization while summing over verb classes.

Now let us consider partially different setting where we sum out over verb classes VC and consider ***distributions per frames classes C*** instead (as

[5] While *People_by_vocation* and *Leadership* are not formally inherited from *People* frame.
[6] Its intuitive inheritance frame *Democracy* is also not defined.

depicted in Figure 6.5). We get certain purification as the verbs *report*, *say*, *show* and *suggest* from the frame *Statement* and the verbs *know*, *believe* and *think* are in the frame *Awareness* are dominating the corresponding frame–verb class $CV - 56$. While the unique verbs (i.e. verb from the cluster that belong to single $FrameNet$ frame) from the verb–class *decide*, $fear$ and *hope* are clustered differently and are no longer dominating. Further, frames *Giving* and *Getting* are dominating frame–verb class $CV - 47$. Cluster $CV - 44$ contains units of the frame *Change_position_on_a_scale* (i.e. verbs *climb*, *decline*, *drop*, $fell$, *grow*, *increase*, *jump* and *rise*). This shows that while purely clustering information based on the similar syntactic–semantic behavior is present in verb–classes the finer abstractions are derived when considering more structured information (i.e. linguistic, semantic). That can be provided as a claim of preferring $FrameNet$ semantic abstractions over other similar hierarchical abstracts like $VerbNet$ in the general semantic properties.

Finally, we will try to supports our already stated claim with regard to the ***constraining of the structural assignment***. We explore this important assumption by taking the most similar structured assignment over entire proposition. Thus we join all clusters of a single proposition in a single element and find its most similar correspondent via ***Levenshtein*** distance.

In the first example we consider a sentence *He believes in what he plays, and he plays superbly.* The example is shown in the Figure 6.6. Under the example sentence you can see its five closest matching correspondents as represented by the difference in their overall frame proposition. We show a proposition in terms of constituency–tree–like representation while the comparison was done only on cluster assignments. The proposition is presented for the purpose of straightforward identification of the arguments of a example predicate. It is very obvious even without consulting $FrameNet$ that all examples are talking about *opinion*. And indeed the frame *Opinion* contains all of the closest verbs : *believe*, $feel$ and *think*; except the verb *know* which is the frame *Awareness*.

Frame classes
He ***believes*** in what he plays , and he plays superbly . ((V-0 (@V-41 (A0-12 (X-53 he) (Y-22 SBJ0)) (VB-50 believe)) (A1-18 (X-92 plays) (Y-46 ADV0in))))
`` You ***feel*** you want one more -- one more at - bat , one more hit , one more game . '' ((V-0 (@V-41 (A0-12 (X-53 you) (Y-22 SBJ0)) (VB-50 feel)) (A1-18 (X-69 want) (Y-92 OBJ0)))) -- [0.0]
`` I ***think*** so . ((V-0 (@V-41 (A0-12 (X-53 i) (Y-22 SBJ0)) (VB-50 think)) (A1-18 (X-93 so) (Y-44 ADV0)))) -- [2.0]
A lot of observers ***think*** so , and , if they 're right , the whole economy as well as the spendthrifts among us could be hurt . ((V-0 (@V-41 (A0-12 (X-53 lot) (Y-22 SBJ0)) (VB-50 think)) (A1-18 (X-93 so) (Y-44 ADV0)))) -- [2.0]
But the nations of Europe and North America have ***decided*** they know better . ((V-0 (@V-41 (A0-12 (X-53 they) (Y-22 SBJ0)) (VB-50 know)) (A1-18 (X-93 better) (Y-44 ADV0)))) -- [2.0]
Some Democrats ***thought*** they might have compromised too much . ((V-0 (@V-41 (A0-12 (X-53 democrats) (Y-22 SBJ0)) (VB-50 think)) (A1-18 (X-69 might) (Y-92 OBJ0)))) -- [4.0]

Figure 6.6: First example of the frame class as defined by the overall structural similarity over induced clusters.

However, if one closely check the example sentence with the verb *know* it is quite possible to suspect that that the actual meaning in this particular case is in the form of belief.

Frame classes
The number of people registered as jobless at the end of October ***declined*** by 900 from September to 78,600 . ((V-0 (@V-67 (@V-66 (@V-57 (A1-3 (X-45 number) (Y-23 SBJ0)) (VB-87 decline)) (A2-32 (X-11 900) (Y-68 EXT0by))) (AMTMP-16 (X-82 september) (Y-58 DIR0from))) (A4-21 (X-10 78,600) (Y-84 DIR0to))))
The average maturity of the taxable funds that Donoghue 's follows ***increased*** by two days in the latest week to 40 days , its longest since August . ((V-0 (@V-67 (@V-66 (@V-57 (A1-3 (X-45 maturity) (Y-23 SBJ0)) (VB-87 increase)) (A2-32 (X-11 days) (Y-68 EXT0by))) (AMTMP-16 (X-82 week) (Y-53 TMP0in))) (A4-21 (X-10 longest) (Y-84 DIR0to)))) -- [1.0]
The DAX ***dropped*** 19.69 points Friday to 1462.93 . ((V-0 (@V-67 (@V-66 (@V-57 (A1-3 (X-45 dax) (Y-23 SBJ0)) (VB-87 drop)) (A2-32 (X-11 points) (Y-68 EXT0))) (AMTMP-14 (X-78 friday) (Y-48 TMP0))) (A4-21 (X-10 1462.93) (Y-84 DIR0to)))) -- [4.0]
The contract ***fell*** five points at the open to 323.85 , the maximum opening move allowed under safeguards adopted by the Merc to stem a market slide . ((V-0 (@V-67 (@V-66 (@V-57 (A1-3 (X-45 contract) (Y-23 SBJ0)) (VB-87 fell)) (A2-32 (X-11 points) (Y-68 EXT0))) (AMTMP-15 (X-83 open) (Y-52 TMP0at))) (A4-21 (X-0 323.85) (Y-84 DIR0to)))) -- [4.0]
Contracting for non - residential buildings ***rose*** 10 % in September to an annualized $ 100.8 billion . ((V-0 (@V-66 (@V-66 (@V-57 (A1-3 (X-45 contracting) (Y-23 SBJ0)) (VB-87 rise)) (A2-32 (X-11 %) (Y-68 EXT0))) (AMTMP-16 (X-82 september) (Y-53 TMP0in))) (A4-22 (X-8 $) (Y-84 DIR0to)))) -- [5.0]
Personal spending ***grew*** 0.2 % in September to a $ 3.526 trillion annual rate , the Commerce Department said . ((V-0 (@V-66 (@V-66 (@V-57 (A1-3 (X-45 spending) (Y-23 SBJ0)) (VB-87 grow)) (A2-32 (X-11 %) (Y-68 EXT0))) (AMTMP-16 (X-82 september) (Y-53 TMP0in))) (A4-22 (X-8 rate) (Y-84 DIR0to)))) -- [5.0]

Figure 6.7: Second example of the frame class as defined by the overall structural similarity over induced clusters.

In the second example we see the instances of the frame $Change\_position\_on-$ $-\_a\_scale$ (see Figure 6.7). This sample frame has a very specific structure and the meaning as all of examples talk about changing the value of the various properties – $Attributes$ to an end value $Final\_value$ or to some degree – $Difference$ in some time $Time$. All of which are predefined by the $FrameNet$ core and non–core frame elements. In our case the structure is preserved specifically because of modeling of the linking. Note that while the structured similarity is present in a great extent the surface realization in both of terms of lexical and syntactic elaboration is partially invariant (e.g. see Figure 6.8).

Our last and third example considers several $Motion$ inheritance frames (see Figure 6.8). Some of the example have very literal while some other extremely metaphorical meaning an their direct correspondence cannot be found in $Frame-$ $-Net$.

Frame classes
`` I personally do n't enjoy seeing players who I remember vividly from their playing days ***running*** about and being gallant about their deficiencies , '' says Roger Angell , New Yorker magazine 's resident baseball sage . ((V-0 (@V-63 (A1-2 (X-46 players) (Y-1 OPRD1OBJ0)) (VB-107 run)) (AMDIR-11 (X-15 about) (Y-56 DIR0))))
Many economic - development officials say the Koch administration 's aggressive approach helped save 5,000 Chase Manhattan Bank jobs from ***moving*** across the Hudson . ((V-0 (@V-63 (A1-2 (X-46 jobs) (Y-1 PMOD1ADV1OBJ0)) (VB-107 move)) (AMDIR-11 (X-15 hudson) (Y-56 LOC0across)))) -- [0.0]
But is he so clever that he has achieved the political equivalent of making water ***run*** uphill ? ((V-0 (@V-63 (A1-2 (X-46 water) (Y-1 OPRD1OBJ0)) (VB-107 run)) (AMDIR-11 (X-15 uphill) (Y-56 DIR0)))) -- [0.0]
The move was interpreted by some economists as a sign that the Fed does n't want the federal funds rate to ***move*** any lower than the 8 3\/4 % at which it has been hovering around during the past week . ((V-0 (@V-63 (A1-2 (X-46 rate) (Y-1 IM1OPRD1OBJ0)) (VB-107 move)) (AMDIR-11 (X-15 lower) (Y-56 DIR0)))) -- [0.0]
Only his factories in Japan and Korea , employing his followers at subsistence wages and producing everything from rifles to ginseng to expensive marble vases , kept the money ***flowing*** westward . ((V-0 (@V-63 (A1-2 (X-46 money) (Y-1 OPRD1OBJ0)) (VB-107 flow)) (AMDIR-11 (X-15 westward) (Y-56 DIR0)))) -- [0.0]
`` The industry has been waiting with bated breath for the machines to ***come*** along , '' says David Niles , president of Eleven Twenty Five Productions Inc. , a New York pioneer in high - definition programming . ((V-0 (@V-63 (A1-2 (X-46 machines) (Y-1 IM1SBJ0)) (VB-107 come)) (AMDIR-11 (X-15 along) (Y-56 DIR0)))) -- [0.0]

Figure 6.8: Third example of the frame class as defined by the overall structural similarity over induced clusters.

6.2 CONLL09

We *evaluate* our model on the data provided by the ***CoNLL-09 shared task*** [9]. The focus of the task was to perform joint learning and inference over syntactic and semantic dependency structures over ***multiple languages***. The semantic relations included, apart from the standard verbal predicates, propositions over other major part-of-speech categories. Motivated by recent works in unsupervised semantic role labeling, we assume the argument identification being provided to us and focus on ***labeling***. Also, the task of argument identification is solved with an extreme precision, with discriminative methods having a cross-language accuracy of over 95%. For the details of the data and the task we encourage the reader to see [9]. Most important is to note that we compare our results to systems which competed in so called $SRL - only$ task. This task focused on semantic role labeling and indeed brought better results that the systems which tried to jointly infer over both syntactic and semantic dependencies.

For evaluation we use standard script from the task $eval09.pl$ [7], but make one important adjustment when comparing our results to multiple other systems (i.e. when comparing to a single system it does not make much of a difference: the order remains while the magnitude slightly varies). Namely, as we do not predict the sense of the verb in our model we exclude it from the evaluation.

The standard way to evaluate semantic propositions is by converting them to ***semantic dependencies*** (i.e. if the verb has n arguments it will have n semantic dependencies). Then the classification task is actually reduced to labeling these dependencies with semantic roles. Additionally, a semantic dependency from a predicate to a virtual $ROOT$ node is created in order to evaluate the

[7] The script is available at `http://ufal.mff.cuni.cz/conll2009-st/eval09.pl`.

Rank	System	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
1	Zhao	**65,36**	**76,99**	69,83	**77,59**	**81,00**	74,46	**77,68**	**65,36**
2	Nugues	64,95	76,76	**70,87**	74,77	**81,00**	**78,79**	72,46	64,95
3	**M1**	61,25	72,19	66,54	69,73	78,67	68,36	73,24	61,25
4	**M2**	58,12	70,48	66,42	54,56	77,20	67,40	70,77	58,12

Table 6.1: Final performance of our system compared with the CoNLL09 top performing systems. Bold numbers represent best results over entire task.

prediction of the verb sense. We simply exclude this additional *ROOT* node in the evaluation script and compare only over semantic roles. Thus if the system has predicted the proposition:

$$\text{verb.01} : \text{A1 A0 AM} - \text{LOC}$$

as a approximation of the correct proposition:

$$\text{verb.02} : \text{A0 A1 AM} - \text{LOC}$$

it will receive labeled precision score of 2/3 in the new metric, while for the standard metric it would receive 2/4 as the verb is misclassified. Using this setting we estimate labeled *F1* score.

As we assume argument identification to be provided to us we must use some of the existing resources which carry that information. Luckily, the $CoNLL09$ shared task has made available competing system outputs on its official web page [8]. Format of the data is partialy described in the official summary paper [9] but as $CoNLL$ is an annual competition that is running for several consecutive years most of the descriptions backtrack to previous years. Thus, reader interested in formating details can start from [9]. After having these data we have to choose one of the competing systems outputs as our argument identifier. As our system is trained on gold arguments with a generative model its performance would be very much bound by the argument identification step. Discriminative approaches can tune their parameters on the development set and even on train set and in that way leverage argument identification and classification with respect to the final output. Thus, we decide to be take the outputs of the best argument indetifier from each language. Consequently, for *Catalan*, *Japanese* and *Spanish* we use argument identification from system *Zhao* and for *Chinese*, *Czech*, *English* and *German* we use from system *Nugues*.

Concerning the optimization of our system, on the first place we have to choose the $split - merge$ step. We do that by evaluating on development set and conclude that the highest possible $split - merge$ step obtainable by our software package (i.e. 7 $split - merge$ steps) is always performing the best for languages with high amount of training data (e.g. $Czech$, $English$) and conversely, one

[8]The outputs are available at `http://ufal.mff.cuni.cz/conll2009-st/eval-data.zip`.

Rank	System	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
1	Merlo	**62,66**	**73,74**	**67,26**	**75,77**	78,66	**69,55**	**73,62**	**62,66**
3	**M1**	61,25	72,19	66,54	69,73	**78,67**	68,36	73,24	61,25
4	**M1-MA**	60,66	71,55	66,48	68,98	77,13	67,54	72,91	60,66
4	**M2**	58,12	70,48	66,42	54,56	77,20	67,40	70,77	58,12

Table 6.2: Final performance of our system compared with $Merlo$ system. MA denotes that the argument boundaries and intermediate structures have been prediceted for our system by $Merlo$ system. Otherwise numbers have just been copied from the Figure 6.9 and thus exactly the same interpretation.

iteration less (i.e. 6 steps) for languages with small amount of training data (e.g. $Catalan$).

As you can see from the Table 6.2 our model performs very close to the best systems which in fact represent state–of–the–art at the time of writing this thesis. Our systems are especially good in $Czech$ having the closest absolute difference with respect to the best results. This is mainly the case as the linguistic theory by which the semantic roles in $Czech$ were annotated – $Functional\ Generative\ Des- -cription$ made an effort to support consistent descriptions across predicates. Oppositely, for $Japanese$ we get substantially lower performance that for the other languages, with respect to the gap toward the best system and also with respect to the variance of our system across languages.

Further, our model $M1$ is slightly outperforming its *linguistically motivated* variant $M2$. But in fact the $M2$ is learning something a bit different which is less oriented toward predicting semantic roles exclusively. Thus our continued analysis will focus on the performance of the $M1$ model.

We further argue that the setting in which we are achieving the results reported in Table 6.1 is quite unfair with regards to our system. We have language–independent, generative and linguistically motivated model with minimum domain–specificity and structured–specificity. By using only dependency path as the source of information our model is seriously suppressed by discriminative models which directly optimize for the task by using millions of features each tuned per language. Thus our performance would be easier to judge if we would compare only to systems which have at least some of the restrictions we impose. Consequently, we find that one of the systems (i.e. $Merlo$ system) competed on the so called $Joint - task$ and has been developed with minimum feature engineering and has been mostly inducing features in both structured and syntactic space. We have already seen the description of this model in the Section 3.1. Specifically, $Merlo$ system is a generative model as well, it uses more features than our model, but still less features than the other systems, meanwhile avoiding manual interventions as possible. The $Joint-task$ implied prediction of both syntactic and semantic dependencies and thus we evaluate our model as well on arguments and intermediate structures (e.g. syntactic dependencies) predicted by $Merlo$ system. We depict the performance in Table 6.2.

We note that our system trained on the structures predicted by $Merlo$ sys-

tem performs much more worse than before. Only indifferent language being *Japanese* – our most problematic language. On the other hand our best performing language *English* also has a big drop in performance and in fact our argument identifier from the $SRL - only$ has lower performance on $Czech$ than the $Merlo$ system. We argue that this is the case because of the reason that $Merlo$ system induces close latent correspondence between syntactic and semantic dependencies and in that way leverages between structures in hidden decision which are then on the output lost [6]. We can support these claims by having that all syntactic parsers make fair number of errors in important linguistic constructions [3] while the trivial and local structures are boost to their performance which is superficially considered as being of a high degree. Furthermore, $Merlo$ system in fact has a best performance for $Czech$ syntactic parsing and second best performance for semantic parsing. Now remind that the $Czech$ theory *Functional Generative Description* intended for roles to be shared across predicates. Then one can make the following claim: *The most successful example of the close correspondence between syntax and semantics on the CoNLL09 shared task and the possible advantages of joint learning is presented by the system Merlo on Czech language.* $Merlo$ system – as it is the only system which considers latent correspondence and $Czech$ language as the treebank annotations are most sound with general properties of the assumed semantic and semantic properties. However, this correspondence is lost on the output [6] and thus we are unable to learn it as good as we are for the consistent error–making parsers as provided by $SRL - only$ task where both the train data and test data come from the same parser – $Malt$ parser [21].

Furthermore, as our model is of generative nature trained only on training data we expect that different kinds of errors in argument identification are very much influential on the overall preposition by remembering that our model will always try to find some meaningful derivation on the cluster assignments . We conclude that the performance of our system is very much comparable with performance of the $Merlo$ system and that some future work of jointly learning both systactic and semantic dependencies with full joint derivation in the model of our nature is the reasonable direction.

7. Future work

We have shown competitive results on modeling varying abstraction of semantics trained jointly with syntactic and lexical information. Out model has a simple and compact form optimizing only training likelihood. To further increase the expressiveness of our model as well as the performance one would have many options. First of all, the training objective could be changed so that model directly optimizes some form of validation error or validation likelihood. One might also what to consider using some other graphical model implementation and thus delve into the technicalities of the problem. Jointly learning full derivation of syntactic and lexical representation of the semantics in a single model is definitely required if the model would be considered for the real–word application. The approach presented is very simple and breaks a lot of independence assumptions of the current approaches to semantic parsing at the same time without using any features and abstracting and encapsulation required information. Standard semi–supervised setting would also bring valuable out–of–domain lexical information and further robustness of syntactic and semantic classes. Also using principles defined in our theses one could try semi–supervised abstractions of linguistic structure over multiple languages. Finally, coupling unsupervised variables from our model with linguistic resources (e.g. $FrameNet$, $VerbNet$) would introduce even more real–word knowledge and surely boost performance.

Conclusion

One of the reasonable approaches in dealing with language processing, at least while the language is seen as a string of tokens, is to combine linguistic structures and powerful structure learning algorithms. The first being the necessary word knowledge in some of its forms and the second being empirical reasoning over obscure, incomplete and noisy data. That kind of an approach is using linguistic structures but threat them as a backbone structure for learning while trying to specify the structures and parameters in order to perform well on the task of interest. We have presented semi-supervised latent variable approach for learning varying levels of semantics. Our model does not use any features while jointly learning syntactic and semantic dependencies as suggested by the linking theory. Our model in its simple form shows good cross–lingual performance without any changes in the model. Further we have shown quite a radical new approach which ignores verb-per-verb assumption, that learns linking compactly in the model and that assumes role fillers to be cross-shared. Most importantly we learned semantic frames with varying levels of abstraction. That gives a hope to the aim of semantic parsing of multilingual free text while its direct abstractions are learned in the task based manner.

Bibliography

[1] Ackema, P., and Schoorlemmer, M. The middle construction and the syntax-semantics interface. *Lingua 93*, 1 (1994), 59 – 90.

[2] Baker, C. F., Fillmore, C. J., and Lowe, J. B. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics - Volume 1* (Stroudsburg, PA, USA, 1998), COLING '98, Association for Computational Linguistics, pp. 86–90.

[3] Bender, E. M., Flickinger, D., Oepen, S., and Zhang, Y. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2011), EMNLP '11, Association for Computational Linguistics, pp. 397–408.

[4] Björkelund, A., Hafdell, L., and Nugues, P. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task* (Stroudsburg, PA, USA, 2009), CoNLL '09, Association for Computational Linguistics, pp. 43–48.

[5] Christodoulopoulos, C., Goldwater, S., and Steedman, M. Two decades of unsupervised pos induction: how far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2010), EMNLP '10, Association for Computational Linguistics, pp. 575–584.

[6] Gesmundo, A., Henderson, J., Merlo, P., and Titov, I. Latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *CoNLL 2009 Shared Task., Conf. on Computational Natural Language Learning* (Boulder, Colorado, USA, 2009), pp. 37–42.

[7] Gildea, D., and Jurafsky, D. Automatic labeling of semantic roles. *Computational Linguistics 28* (2001), 245–288.

[8] Grenager, T., and Manning, C. D. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2006), EMNLP '06, Association for Computational Linguistics, pp. 1–8.

[9] Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. The conll-2009 shared task: syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task* (Stroudsburg, PA, USA, 2009), CoNLL '09, Association for Computational Linguistics, pp. 1–18.

[10] Henderson, J., and Titov, I. Incremental sigmoid belief networks for grammar learning. *Journal of Machine Learning Research (JMLR) 11* (2010), 3541–3570.

[11] Jurafsky, D., and Martin, J. *Speech and language processing, 2nd edition.* Prentice Hall, 2008.

[12] Lang, J., and Lapata, M. Unsupervised induction of semantic roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2010), HLT '10, Association for Computational Linguistics, pp. 939–947.

[13] Lang, J., and Lapata, M. Unsupervised semantic role induction with graph partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Stroudsburg, PA, USA, 2011), EMNLP '11, Association for Computational Linguistics, pp. 1320–1331.

[14] Levin, B. *English Verb Classes And Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, 1993.

[15] Liu, D., and Gildea, D. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics* (Stroudsburg, PA, USA, 2010), COLING '10, Association for Computational Linguistics, pp. 716–724.

[16] Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S. Semantic role labeling: an introduction to the special issue. *Comput. Linguist. 34*, 2 (June 2008), 145–159.

[17] Matsuzaki, T., Miyao, Y., and Tsujii, J. Probabilistic cfg with latent annotations, 2005.

[18] Naradowsky, J., Riedel, S., and Smith, D. A. Improving nlp through marginalization of hidden syntactic structure. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP '12)* (Jeju, Korea, July 2012), Association for Computational Linguistics.

[19] Naseem, T., Barzilay, R., and Globerson, A. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Jeju Island, Korea, July 2012), Association for Computational Linguistics, pp. 629–637.

[20] Ng, A. Y., and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, 2001.

[21] Nivre, J., and Hall, J. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories* (2005), pp. 13–95.

[22] Palmer, M., Gildea, D., and Kingsbury, P. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist. 31*, 1 (Mar. 2005), 71–106.

[23] Petrov, S., Barrett, L., Thibaux, R., and Klein, D. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (Sydney, Australia, July 2006), Association for Computational Linguistics, pp. 433–440.

[24] Poon, H., and Domingos, P. Unsupervised Semantic Parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Singapore, Aug. 2009), Association for Computational Linguistics, pp. 1–10.

[25] Punyakanok, V., Roth, D., and Yih, W.-t. The importance of syntactic parsing and inference in semantic role labeling. *Comput. Linguist. 34*, 2 (June 2008), 257–287.

[26] Sammons, M., Vydiswaran, V., Vieira, T., Johri, N., Chang, M., Goldwasser, D., Srikumar, V., Kundu, G., Tu, Y., Small, K., Rule, J., Do, Q., and Roth, D. Relation alignment for textual entailment recognition. In *TAC* (2009).

[27] Schuler, K. K. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon.* PhD thesis, University of Pennsylvania, 2006.

[28] Shen, D., and Lapata, M. Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (Prague, Czech Republic, June 2007), Association for Computational Linguistics, pp. 12–21.

[29] Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1* (Stroudsburg, PA, USA, 2003), ACL '03, Association for Computational Linguistics, pp. 8–15.

[30] Titov, I., and Klementiev, A. A bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland, Oregon, USA, June 2011), Association for Computational Linguistics, pp. 1445–1455.

[31] Titov, I., and Klementiev, A. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics* (Avignon, France, April 2012).

[32] Titov, I., and Klementiev, A. Crosslingual induction of semantic roles. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (Jeju Island, South Korea, July 2012), Association for Computational Linguistics.

[33] Wu, D., and Fung, P. Semantic roles for smt: a hybrid two-pass model. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (Stroudsburg, PA, USA, 2009), NAACL-Short '09, Association for Computational Linguistics, pp. 13–16.

[34] Zhao, H., Chen, W., Kazama, J., Uchimoto, K., and Torisawa, K. Multilingual dependency learning: exploiting rich features for tagging syntactic and semantic dependencies. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task* (Stroudsburg, PA, USA, 2009), CoNLL '09, Association for Computational Linguistics, pp. 61–66.

A. Basic notation

In Figure A.1 we provide short description of the notation used in generative and grammar models.

Notation for Generative models

$\phi_f \sim D^{(F)}$ – *univariate distribution of variable* f *in the dataset* D *by maximum likelihood estimation (MLE)*

$\phi_f \sim D^{(X),(Y),(F)}$ – *conditional distribution of variables* x *and* y *given the variable* f *in the dataset* D *by MLE*

$x, y \sim \phi_f$ – *sampling* x *and* y *for the given* f (*note that we assume non ambiguity among* ϕ *distributions according to the type of input arguments* – f *in this case*)

Notation for Grammars

$S \rightarrow F$ – *variable* S *generates variable* F; *where each variable can take any number of symbols with respect to the underlying variable type* (*e.g. if* F *represents unsupervised variable then it has one subsymbol*)

$S \rightarrow F[X]$ – *variable* S *generates variable* F; *where each variable can take any number of symbols with respect to the underlying variable type and furthermore each symbol is specified with an integer subsymbol* (*e.g. if* F *takes symbol is* $A1$ *it specified form could be* $A1 - 5$)

Figure A.1: Notation used in generative and grammar models